

# پایتون برای برنامه‌نویسان (کتاب الکترونیکی)

---

---

تألیف:

پل دایتل و هاروی دایتل

مترجم:

سیامک وطنی



فن آوری نوین

---

---

شماره کتابشناسی ملی	: ۷۶۷۳۵۹۳
شابک	: ۹۷۸-۶۲۲-۷۳۹۳-۳۹-۲ ریال : ۱۹۹۰۰۰۰
سرشناسه	: دیتل، پل ج.، ۱۹۶۹ - م. <b>Deitel, Paul J</b>
عنوان و نام پدیدآور	: پایتون برای برنامه‌نویسان منابع الکترونیکی: کتاب // پل دایتل، هاروی دایتل؛ مترجم سیامک وطنی.
مشخصات نشر	: بابل: فناوری نوین، ۱۴۰۰.
مشخصات ظاهری	: ۱ لوح فشرده (۷۲۳ ص): مصور، جدول؛ ۱۲ س.م.
وضعیت فهرست نویسی	: فیپا
یادداشت	: عنوان اصلی: <b>Python for programmers : with big data and artificial intelligence case studies, 2019.</b>
نوع منبع الکترونیکی	: فایل متنی (PDF).
ویراست دیگر از اثر در قالب دیگر رسانه	: پایتون برای برنامه‌نویسان [کتاب]
شناسه افزوده	: دیتل، هاروی ام.، ۱۹۶۵ - م.
شناسه افزوده	: <b>Deitel, Harvey M</b>
شناسه افزوده	: وطنی، سیامک، ۱۳۴۹ - مترجم
موضوع	: پایتون (زبان برنامه‌نویسی کامپیوتر)
موضوع	: <b>Python (Computer program language)</b>
زده بندی کنگره	: ۷۶/۷۳QA
زده بندی دیویی	: ۰۰۵/۱۳۳

@fanavarienovinpub

تلفن: ۰۱۱-۳۲۲۵۶۶۸۷

بابل، کد پستی ۴۷۱۶۷-۷۳۴۴۸

فن آوری نوین

### پایتون برای برنامه‌نویسان

مترجم: سیامک وطنی

نوبت چاپ: چاپ اول

سال چاپ: بهار ۱۴۰۰

شمارگان: ۲۰۰

قیمت: ۱۹۹۰۰۰ تومان

نام چاپخانه و صحافی: دفتر فنی سورنا

شابک: ۹۷۸-۶۲۲-۷۳۹۳-۳۹-۲

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

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

**پخش و فروش کتاب‌های چاپی: تهران، تلفن ۰۲۱-۶۶۴۰۰۱۴۴-۶۶۴۰۰۲۲۰**

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

## فهرست مطالب

### فصل اول: مقدمه‌ای بر کامپیوترها و پایتون

- ۱-۱. مقدمه ..... ۲۵
- ۱-۲. مرور سریع اصول فناوری شیء ..... ۲۶
- ۱-۳. پایتون ..... ۲۷
- ۱-۴. کتابخانه‌ها ..... ۳۱
- ۱-۴-۱. کتابخانه‌های استاندارد پایتون ..... ۳۳
- ۱-۴-۲. کتابخانه‌های علوم داده ..... ۳۴
- ۱-۵. اجرای آزمایشی با استفاده از IPython و  
Jupyter Notebooks ..... ۳۶
- ۱-۵-۱. استفاده از IPython در وضعیت محاوره‌ای  
به‌عنوان یک ماشین حساب ..... ۳۷
- ۱-۵-۲. اجرای یک برنامه پایتون با استفاده از  
IPython ..... ۳۹
- ۱-۵-۳. نوشتن و اجرای کد در Jupyter  
Notebook ..... ۴۳
- ۱-۶. ابر و اینترنت اشیا ..... ۴۶
- ۱-۶-۱. ابر ..... ۴۶
- ۱-۶-۲. اینترنت اشیا ..... ۴۷
- ۱-۷. کلان داده‌ها تا چه حد بزرگ هستند؟ ..... ۱۶
- ۱-۷-۱. تجزیه و تحلیل کلان داده ..... ۴۸
- ۱-۷-۲. علوم داده و کلان داده نتایج متفاوتی خلق  
می‌کنند: موارد کاربرد ..... ۵۷
- ۱-۸. مورد مطالعاتی - یک برنامه تلفن همراه مبتنی بر  
کلان داده ..... ۱۷

- ۱-۹. مقدمه‌ای بر علوم داده: هوش مصنوعی فصل اشتراک  
علوم کامپیوتر و علوم داده ..... ۵۹
- ۱-۱۰. جمع‌بندی ..... ۶۳

### فصل دوم: مقدمه‌ای بر برنامه‌نویسی پایتون

- ۲-۱. مقدمه ..... ۶۴
- ۲-۲. متغیرها و دستورات انتساب ..... ۶۵
- ۲-۳. ریاضیات ..... ۶۶
- ۲-۴. تابع print و مقدمه‌ای بر رشته‌های محصور با  
کاراکتر نقل قول تکی، دوتایی ..... ۷۰
- ۲-۵. رشته‌هایی با کاراکتر نقل قول سه‌تایی ..... ۷۲
- ۲-۶. گرفتن ورودی از کاربر ..... ۷۳

۲-۷. تصمیم‌گیری: دستور if و عملگرهای

- مقایسه‌ای ..... ۷۵
- ۲-۸. اشیا و داده‌ها با نوع پویا ..... ۸۱
- ۲-۹. مقدمه‌ای بر علوم داده: آمار توصیفی پایه ..... ۸۲
- ۲-۱۰. جمع‌بندی ..... ۸۵

### فصل سوم: دستورات کنترلی

- ۳-۱. مقدمه ..... ۸۷
- ۳-۲. دستورات کنترلی ..... ۸۷
- ۳-۳. دستور if ..... ۸۸
- ۳-۴. دستوره‌های if else و if elif else ..... ۹۰
- ۳-۵. دستور while ..... ۹۳
- ۳-۶. دستور for ..... ۹۳
- ۳-۶-۱. Iteratorها، لیست‌ها و Iterableها ..... ۹۵
- ۳-۶-۲. تابع از پیش تعریف‌شده range ..... ۹۵
- ۳-۷. دستورات انتساب کوتاه شده ..... ۹۶
- ۳-۸. تکرارهای کنترل شده روی دنباله‌ها: رشته‌های  
فرمت شده ..... ۹۷
- ۳-۹. حلقه‌هایی که با نشانه خاص کنترل می‌شوند ..... ۹۸
- ۳-۱۰. بررسی عمیق‌تر تابع range ..... ۱۰۰
- ۳-۱۱. استفاده از نوع decimal برای مبالغ پولی ..... ۱۰۱
- ۳-۱۲. دستورات break و continue ..... ۱۰۵
- ۳-۱۳. عملگرهای منطقی and و or و not ..... ۱۰۵
- ۳-۱۴. مقدمه‌ای بر علوم داده: سنجش شاخص مرکزی  
- میانگین، مد و میانگین ..... ۱۰۸
- ۳-۱۵. جمع‌بندی ..... ۱۱۰

### فصل چهارم: توابع

- ۴-۱. مقدمه ..... ۱۱۲
- ۴-۲. تعریف توابع ..... ۱۱۳
- ۴-۳. توابع با چندین پارامتر ..... ۱۱۶
- ۴-۴. مولد اعداد تصادفی ..... ۱۱۷
- ۴-۵. مورد مطالعاتی: یک بازی شانس ..... ۱۲۰
- ۴-۶. کتابخانه استاندارد پایتون ..... ۱۲۴
- ۴-۷. توابع ماژول math ..... ۱۲۶
- ۴-۸. استفاده از tab برای تکمیل کد در  
IPYTHON ..... ۱۲۶
- ۴-۹. مقادیر پیش‌فرض آرگومان‌ها ..... ۱۲۸
- ۴-۱۰. آرگومان‌های کلیدی ..... ۱۲۹
- ۴-۱۱. لیستی دلخواه از آرگومان‌ها ..... ۱۲۹
- ۴-۱۲. متدها: توابعی که به اشیا تعلق دارند ..... ۱۳۱
- ۴-۱۳. قوانین حوزه ..... ۱۳۱
- ۴-۱۴. تکه‌های عمیق‌تر به import ..... ۱۳۳
- ۴-۱۵. ارسال آرگومان‌ها به توابع: تکه‌های عمیق‌تر .. ۱۳۵

## فصل هفتم: برنامه‌نویسی آرایه‌ها با Numpy

۲۱۷	۷-۱. مقدمه
۲۱۸	۷-۲. ایجاد آرایه با استفاده از داده‌های موجود
۲۱۹	۷-۳. ویژگی‌های آرایه
۲۲۱	۷-۴. ایجاد آرایه با مقادیر خاص
۲۲۱	۷-۵. ایجاد آرایه با محدوده‌ها
	۷-۶. کارایی لیست در قیاس با آرایه: مقدمه‌ای بر
۲۲۳	timeit/
۲۲۵	۷-۷. عملگرهای آرایه
۲۲۷	۷-۸. متدهای محاسباتی numpy
۲۲۸	۷-۹. توابع جامع
۲۳۰	۷-۱۰. ایندکس و برش
۲۳۱	۷-۱۱. view: کپی‌های سطحی
۲۳۳	۷-۱۲. کپی‌های عمیق
۳۲۴	۷-۱۳. ترانزاده و تغییر شکل
	۷-۱۴. مقدمه‌ای بر علوم داده: Series و
۲۳۶	DataFrame در کتابخانه pandas
۲۳۸	۷-۱۴-۱. pandas در Series
۲۴۲	۷-۱۴-۲. DataFrame‌ها
۲۵۰	۷-۱۵. جمع‌بندی

## فصل هشتم: رشته‌ها: نگاهی عمیق‌تر

۲۵۴	۸-۱. مقدمه
۲۵۵	۸-۲. فرمت دادن به رشته‌ها
۲۵۵	۸-۲-۱. نمایش نوع داده
۲۵۷	۸-۲-۲. عرض فیلد و تراز
۲۰۷	۸-۲-۳. فرمت کردن اعداد
۲۵۸	۸-۲-۴. متدهای فرمت کردن رشته
۲۵۹	۸-۳. الصاق و تکرار رشته‌ها
۲۶۰	۸-۴. حذف فضای خالی از رشته‌ها
۲۶۰	۸-۵. تغییر بزرگی و یا کوچکی کاراکترها
۲۶۱	۸-۶. عملگرهای مقایسه در رشته
۲۶۱	۸-۷. جستجو برای زیررشته‌ها
۲۶۳	۸-۸. جایگزین کردن زیررشته‌ها
۲۶۳	۸-۹. تفکیک کردن و اتصال رشته‌ها
۲۶۶	۸-۱۰. متدهای مربوط به کاراکترها
۲۶۷	۸-۱۱. رشته‌های خام
۲۶۷	۸-۱۲. مقدمه‌ای بر عبارات منظم
۲۶۹	۸-۱۲-۱. re ماژول و تابع fullmatch
۲۷۲	۸-۱۲-۲. جایگذاری زیررشته‌ها و تفکیک رشته‌ها
۲۷۳	۸-۱۲-۳. سایر توابع جستجو: دسترسی به مطابقت‌ها
	۸-۱۳. مقدمه‌ای بر علوم داده: Panda، عبارات منظم
۲۷۶	و داده‌کاوی
۲۸۱	۸-۱۴. جمع‌بندی

## فصل نهم: فایل‌ها و استنادها

۲۸۳	۹-۱. مقدمه
۲۸۵	۹-۲. فایل‌ها
۲۸۶	۹-۳. پردازش فایل‌های متنی
	۹-۳-۱. نوشتن در یک فایل متنی: معرفی دستور
۲۸۶	with
۲۸۷	۹-۳-۲. خواندن داده از یک فایل متنی

۱۳۸	۴-۱۶. بازگشتی
۱۴۱	۴-۱۷. برنامه‌نویسی به سبک تابعی
۱۴۳	۴-۱۸. مقدمه‌ای بر علوم داده: سنجش پراکندگی
۱۴۵	۴-۱۹. جمع‌بندی

## فصل پنجم: دنباله‌ها: لیست‌ها و چندتایی‌ها

۱۴۸	۵-۱. مقدمه
۱۴۹	۵-۲. لیست‌ها
۱۵۳	۵-۳. چندتایی‌ها
۱۵۵	۵-۴. تفکیک اعضای دنباله
۱۵۷	۵-۵. استخراج یک زیر دنباله
۱۵۹	۵-۶. دستور DEL
۱۶۱	۵-۷. ارسال لیست‌ها به توابع
۱۶۲	۵-۸. مرتب کردن لیست‌ها
۱۶۳	۵-۹. جست‌وجو در دنباله‌ها
۱۶۵	۵-۱۰. سایر متدهای لیست
۱۶۷	۵-۱۱. شبیه‌سازی پشته‌ها با استفاده از لیست‌ها
۱۶۸	۵-۱۲. ایجاد لیست با روش خلاصه
۱۶۹	۵-۱۳. مولد عبارات
۱۷۰	۵-۱۴. فیلتر، نگاشت و کاهش‌دهنده‌ها
۱۷۴	۵-۱۵. لیست‌های دوبعدی
۱۷۴	۵-۱۶. مقدمه‌ای بر علوم داده: شبیه‌سازی و رسم نمودار
۱۷۷	۵-۱۶-۱. نمودار نمونه در پرتاب یک تاس برای ۱۷۷،
۱۷۷	۶۰۰۰ و ۶۰۰۰۰ پرتاب
۱۷۷	۵-۱۶-۲. نمودار تکرار و درصد اعداد در پرتاب
۱۷۹	تاس
۱۸۸	۵-۱۷. جمع‌بندی

## فصل ششم: دیکشنری‌ها و مجموعه‌ها

۱۹۱	۶-۱. مقدمه
۱۹۱	۶-۲. دیکشنری‌ها
۱۹۲	۶-۲-۱. ایجاد دیکشنری
۱۹۳	۶-۲-۲. پیمایش یک دیکشنری
۱۹۴	۶-۲-۳. عملیات پایه دیکشنری
۱۹۵	۶-۲-۴. متدهای keys و values در دیکشنری
۱۹۶	۶-۲-۵. مقایسه در دیکشنری
۱۹۷	۶-۲-۶. مثال: دیکشنری نمرات دانشجو
۱۹۸	۶-۲-۷. مثال: شمارش کلمات
۲۰۰	۶-۲-۸. متد update در دیکشنری
۲۰۱	۶-۲-۹. روش مختصر ایجاد دیکشنری
۲۰۲	۶-۳. مجموعه‌ها
۲۰۳	۶-۳-۱. مقایسه مجموعه‌ها
۲۰۵	۶-۳-۲. عملیات ریاضی مجموعه‌ها
۲۰۵	۶-۳-۳. عملیات و متدهای مربوط به مجموعه‌های
۲۰۶	تغییرپذیر
۲۰۷	۶-۳-۴. روش مختصر ایجاد مجموعه‌ها
۲۰۷	۶-۴. مقدمه‌ای بر علوم داده: نمودارهای پویا
۲۰۸	۶-۴-۱. طرز کار نمودارهای دینامیکی
۲۱۱	۶-۴-۲. پیاده‌سازی دیاگرام پویا
۲۱۵	۶-۵. جمع‌بندی

۱۰-۶	مورد مطالعاتی: شبیه‌سازی برزیدن ورق و دادن ورق	۳۳۴
۱۰-۶-۱	تست آزمون - محور کلاس‌های Card و DeckOfCards	۳۳۴
۱۰-۶-۲	کلاس Card - مقدمه‌ای بر خصلت‌های کلاس	۳۳۶
۱۰-۶-۳	کلاس DeckOfCards	۳۳۹
۱۰-۶-۴	نمایش تصاویر کارت با Matplotlib	۳۴۱
۱۰-۷	وراثت: کلاس‌های پایه و زیر کلاس‌ها	۳۴۳
۱۰-۸	ایجاد سلسله‌مراتب وراثت؛ مقدمه‌ای بر پلی-مورفسم	۳۴۶
۱۰-۸-۱	کلاس پایه CommissionEmployee	۳۴۶
۱۰-۸-۲	زیر کلاس پایه SalariedCommissionEmployee	۳۴۹
۱۰-۸-۳	پایه‌گذاری SalariedCommissionEmployee و CommissionEmployee به روش چندریختی	۳۵۳
۱۰-۸-۴	تکاتی در مورد برنامه‌نویسی شیء‌گرا و برنامه‌نویسی مبتنی بر شیء	۳۵۴
۱۰-۹	Duck Typing و پلی‌مورفسم	۳۵۴
۱۰-۱۰	اورلود کردن عملگرها	۳۵۶
۱۰-۱۰-۱	تست آزمون - محور کلاس Complex	۳۵۷
۱۰-۱۰-۲	تعریف کلاس Complex	۳۵۸
۱۰-۱۱	سلسله‌مراتب کلاس استثناها و استثناهای سفارشی	۳۶۰
۱۰-۱۲	چندتایی‌های نام‌گذاری شده	۳۶۱
۱۰-۱۳	معرفی مختصر کلاس‌های داده جدید در پایتون ۳.۷	۳۶۲
۱۰-۱۳-۱	ایجاد کلاس داده Card	۳۶۳
۱۰-۱۳-۲	استفاده از کلاس داده Card	۳۶۶
۱۰-۱۳-۳	مزایای کلاس داده نسبت به چندتایی نام-گذاری شده	۳۶۸
۱۰-۱۳-۴	مزایای کلاس داده نسبت به کلاس‌های سنتی	۳۶۸
۱۰-۱۴	اجرای تست با Docstring و doctest	۳۶۹
۱۰-۱۵	فضاهای نام و حوزه	۳۷۳
۱۰-۱۶	مقدمه‌ای بر علوم داده: سری زمانی و رگرسیون خطی ساده	۳۷۶
۱۰-۱۷	جمع‌بندی	۳۸۵

۹-۴	به‌روزرسانی فایل‌های متنی	۲۹۸
۹-۵	سریالایز با JSON	۲۹۰
۹-۶	تمرکز بر روی امنیت: سریالایز و دی‌سریالایز با pickle	۲۹۳
۹-۷	مطالب بیش‌تر در مورد فایل‌ها	۲۹۴
۹-۸	مدیریت استثناها	۲۹۵
۹-۸-۱	تقسیم بر صفر و ورودی نامعتبر	۲۹۵
۹-۸-۲	دستورهای try	۲۹۶
۹-۸-۳	گرفتن چندین استثنا در یک except	۲۹۹
۹-۸-۴	یک تابع یا متد چه استثنائاتی را ایجاد می‌کنند؟	۳۰۰
۹-۸-۵	چه کدی باید در بلاک try قرار گیرد؟	۳۰۰
۹-۹	قسمت FINALLY	۳۰۰
۹-۱۰	ایجاد یک استثنا به‌طور صریح	۳۰۳
۹-۱۱	Stack Unwinding و Traceback‌ها	۳۰۳
۹-۱۲	مقدمه‌ای بر علوم داده: کار با فایل‌های CSV	۳۰۵
۹-۱۲-۱	ملازول CSV در کتابخانه استاندارد پایتون	۳۰۵
۹-۱۲-۲	خواندن فایل‌های CSV و وارد کردن داده-های آن‌ها به DataFrame‌های Pandas	۳۰۸
۹-۱۲-۳	خواندن دیتاست فاجعه تایتانیک	۳۰۹
۹-۱۲-۴	تحلیل ساده داده‌ها با دیتاست فاجعه تایتانیک	۳۱۱
۹-۱۲-۵	هیستوگرام سن مسافر	۳۱۲
۹-۱۳	جمع‌بندی	۳۱۳
<b>فصل دهم: برنامه‌نویسی شیء‌گرا</b>		
۱۰-۱	مقدمه	۳۱۷
۱۰-۲	کلاس سفارشی Account	۳۲۰
۱۰-۲-۱	تست آزمون - محور کلاس Account	۳۲۰
۱۰-۲-۲	تعریف کلاس Account	۳۲۱
۱۰-۲-۳	ترکیب: ارجاع به اشیا به‌عنوان عضوی از کلاس‌ها	۳۲۳
۱۰-۳	کنترل دسترسی به خصلت‌ها	۳۲۴
۱۰-۴	ویژگی‌ها برای دسترسی به داده	۳۲۵
۱۰-۴-۱	تست آزمون - محور کلاس Time	۳۲۵
۱۰-۴-۲	تعریف کلاس Time	۳۲۷
۱۰-۴-۳	تکاتی در مورد طراحی کلاس Time	۳۳۱
۱۰-۵	شبیه‌سازی خصلت‌های خصوصی	۳۳۲

## فصل یازدهم: پردازش زبان طبیعی (NLP)

۱۱-۱	مقدمه	۳۸۸
۱۱-۲	TEXTBLOB	۳۹۰
۱۱-۲-۱	ایجاد یک TEXTBLOB	۳۹۲
۱۱-۲-۲	تفکیک متن به جملات و کلمات	۳۹۲
۱۱-۲-۳	برچسب‌گذاری اجزای گفتار (POS)	۳۹۳
۱۱-۲-۴	استخراج عبارات‌های اسمی	۳۹۴
۱۱-۲-۵	تجزیه و تحلیل احساسات با تحلیل گر پیش فرض TextBlob	۳۹۴
۱۱-۲-۶	تجزیه و تحلیل احساسات با NaiveBayesAnalyzer	۳۹۶
۱۱-۲-۷	شناسایی و ترجمه زبان	۳۹۷
۱۱-۲-۸	عطف: جمع بستن و مفرد سازی	۳۹۸
۱۱-۲-۹	بررسی و تصحیح املاء	۳۹۹
۱۱-۲-۱۰	نرمال‌سازی: ریشه‌یابی و بن‌واژه‌سازی	۳۹۹
۱۱-۲-۱۱	شمارش کلمات	۴۰۰
۱۱-۲-۱۲	معنای کلمه، مترادف و متضاد آن با WordNet	۴۰۱
۱۱-۲-۱۳	حذف ایست واژه‌ها	۴۰۲
۱۱-۲-۱۴	n-grams	۴۰۴
۱۱-۳	رسم نمودار تکرار کلمات با نمودارهای میله‌ای و ابر کلمات	۴۰۵
۱۱-۳-۱	نمودار تکرار کلمات با Pandas	۴۰۵
۱۱-۳-۲	رسم نمودار تکرار کلمات با نمودارهای میله‌ای و ابر کلمات	۴۰۸
۱۱-۴	ارزیابی خوانایی با Textastic	۴۱۱
۱۱-۵	شناسایی موجودیت نام‌دار با spaCy	۴۱۳
۱۱-۶	تشخیص تشابه با spaCy	۴۱۵
۱۱-۷	سایر کتابخانه‌ها و ابزارهای NLP	۴۱۶
۱۱-۸	کاربردهای یادگیری ماشین و یادگیری عمیق زبان طبیعی	۴۱۷
۱۱-۹	دیتاست‌های زبان طبیعی	۴۱۷
۱۱-۱۰	جمع‌بندی	۴۱۸

## فصل دوازدهم: داده‌کاوی توئیتر

۱۲-۱	مقدمه	۴۲۱
۱۲-۲	بررسی اجمالی API‌های توئیتر	۴۲۴
۱۲-۳	ایجاد یک اکانت توئیتر	۴۲۶
۱۲-۴	دریافت احراز هویت توئیتر - ایجاد یک app	۴۲۶
۱۲-۵	توئیتر چیست؟	۴۲۸
۱۲-۶	Tweepy	۴۳۲
۱۲-۷	احراز هویت توئیتر از طریق Tweepy	۴۳۳
۱۲-۸	به دست آوردن اطلاعات درباره یک اکانت توئیتر	۴۳۵
۱۲-۹	مقدمه‌ای بر کورس‌های Tweepy: تعیین فالورها و دوستان یک اکانت	۴۳۷
۱۲-۹-۱	تعیین فالورهای یک اکانت	۴۳۷
۱۲-۹-۲	یک اکانت خاص چه کسی را فالو می‌کند؟	۴۳۹
۱۲-۹-۳	به دست آوردن توئیتهای اخیر کاربر	۴۴۰
۱۲-۱۰	جستجوی توئیتهای اخیر	۴۴۱

۱۲-۱۱	تعیین روندها و موضوعات طرفدار:	
API روندها	۴۴۴	
۱۲-۱۱-۱	مکان‌هایی با موضوعات طرفدار	۴۴۴
۱۲-۱۱-۲	تعیین موضوعات طرفدار	۴۴۵
۱۲-۱۱-۳	ایجاد یک ابر کلمه برای موضوعات طرفدار	۴۴۷
۱۲-۱۲	پاک‌سازی و پیش‌پردازش توئیتهای برای تجزیه و تحلیل	۴۴۸
۱۲-۱۳	API استریمینگ در توئیتر	۴۵۰
۱۲-۱۳-۱	ایجاد زیر کلاس StreamListener	۴۵۰
۱۲-۱۳-۲	مقداردهی کردن اولیه جریان پردازش	۴۵۳
۱۲-۱۴	تحلیل احساسات توئیتهای	۴۵۵
۱۲-۱۵	کدبندی جغرافیایی و نقشه جغرافیایی	۴۵۹
۱۲-۱۵-۱	به دست آوردن توئیتهای نقشه‌یابی	۴۶۱
۱۲-۱۵-۲	توابع سودمند در tweetutilities.py	۴۶۵
۱۲-۱۵-۳	کلاس LocationListener	۴۶۷
۱۲-۱۶	راه‌های ذخیره کردن توئیتهای	۴۶۹
۱۲-۱۷	توئیتر و سری زمانی	۴۶۹
۱۲-۱۸	جمع‌بندی	۴۶۹

## فصل سیزدهم: محاسبات شناختی و واتسون آی‌بی‌ام

۱۳-۱	مقدمه	۴۷۲
۱۳-۲	اکانت ابر آی‌بی‌ام و کنسول ابر	۴۷۳
۱۳-۳	سرویس‌های واتسون	۴۷۴
۱۳-۴	سایر ابزار و سرویس‌ها	۴۷۸
۱۳-۵	SDK ابر پایتون برای توسعه‌دهنده‌ها	۴۸۰
۱۳-۶	مورد مطالعاتی: برنامه مترجم سفری	۴۸۱
۱۳-۶-۱	قبل از اجرای برنامه	۴۸۲
۱۳-۶-۲	تست موردی برنامه	۴۸۳
۱۳-۶-۲	بررسی کامل اسکریپت SimpleLanguageTranslator.py	۴۸۵
۱۳-۷	منابع واتسون	۴۹۷
۱۳-۸	جمع‌بندی	۴۹۹

## فصل چهاردهم: یادگیری ماشینی: طبقه‌بندی، رگرسیون و خوشه‌بندی

۱۴-۱	مقدمه	۵۰۲
۱۴-۱-۱	Scikit-Learn	۵۰۳
۱۴-۱-۲	انواع یادگیری ماشین	۵۰۴
۱۴-۱-۳	دیتاست‌های همراه با Scikit-Learn	۵۰۷
۱۴-۱-۴	مراحل یک مطالعه معمول در علوم داده	۵۰۸
۱۴-۲	مطالعه موردی: طبقه‌بندی با k نزدیک‌ترین همسایگی و دیتاست Digits, بخش ۱	۵۰۸
۱۴-۲-۱	الگوریتم k نزدیک‌ترین همسایگی	۵۱۰
۱۴-۲-۲	بارگذاری دیتاست	۵۱۲
۱۴-۲-۳	رسم نمودار داده‌ها	۵۱۶
۱۴-۲-۴	تقسیم داده برای آموزش و تست	۵۱۸
۱۴-۲-۵	ایجاد مدل	۵۱۹
۱۴-۲-۶	آموزش مدل	۵۲۰
۱۴-۲-۷	پیش‌بینی کلاس‌های رقمی	۵۲۰

۱۵-۶. شبکه‌های عصبی کانولوشن برای بینایی ماشین؛ طبقه‌بندی چندگانه با دیتاست‌های MNIST ..... ۵۸۹

۱۵-۶-۱. بارگذاری دیتاست MNIST ..... ۵۹۱

۱۵-۶-۲. بررسی داده ..... ۵۹۲

۱۵-۶-۳. آماده‌سازی داده ..... ۵۹۴

۱۵-۶-۴. ایجاد یک شبکه عصبی ..... ۵۹۶

۱۵-۶-۵. آموزش و ارزیابی مدل ..... ۶۰۷

۱۵-۶-۶. ذخیره کردن و بارگذاری مدل ..... ۶۱۳

۱۵-۶-۷. نمودار آموزش شبکه عصبی با TensorBoard ..... ۶۱۳

۱۵-۷. ConvnetJS: آموزش و گرافیک یادگیری عمیق مبتنی بر مرورگر ..... ۶۱۸

۱۵-۸. شبکه‌های عصبی بازگشتی برای دنباله‌ها؛ تجزیه و تحلیل احساسات با دیتاست IMDb ..... ۶۱۸

۱۵-۸-۱. بارگذاری دیتاست مربوط به نقد و بررسی فیلم‌های IMDb ..... ۶۱۹

۱۵-۸-۲. بررسی داده ..... ۶۲۰

۱۵-۸-۳. آماده‌سازی داده ..... ۶۲۳

۱۵-۸-۴. ایجاد شبکه عصبی ..... ۶۲۴

۱۵-۸-۵. آموزش و ارزیابی مدل ..... ۶۲۷

۱۵-۹. تنظیم مدل‌های یادگیری عمیق ..... ۶۲۸

۱۵-۱۰. مدل‌های از قبل آموزش دیده شبکه‌های کانولوشن برای ImageNet ..... ۶۲۹

۱۵-۱۱. جمع‌بندی ..... ۶۳۰

## فصل شانزدهم: کلان داده: هادوب، اسپارک،

### NoSQL و اینترنت اشیا

۱۶-۱. مقدمه ..... ۶۳۵

۱۶-۲. پایگاه داده‌های رابطه‌ای و زبان پرس‌وجوی ساخت یافته (SQL) ..... ۶۴۰

۱۶-۲-۱. پایگاه داده books ..... ۶۴۲

۱۶-۲-۲. کوئری‌های SELECT ..... ۶۴۷

۱۶-۲-۳. قسمت WHERE ..... ۶۴۷

۱۶-۲-۴. قسمت ORDER BY ..... ۶۴۸

۱۶-۲-۵. ادغام داده‌ها از چند جداول: INNER JOIN ..... ۶۵۰

۱۶-۲-۶. دستور INSERT INTO ..... ۶۵۱

۱۶-۲-۷. دستور UPDATE ..... ۶۵۲

۱۶-۲-۸. دستور DELETE FROM ..... ۶۵۳

۱۶-۳. پایگاه داده NoSQL و NewsQL برای کلان داده: بررسی کوتاه ..... ۶۵۳

۱۶-۳-۱. پایگاه داده‌های NoSQL از نوع کلید-مقدار ..... ۶۵۴

۱۶-۳-۲. پایگاه داده‌های NoSQL از نوع اسناد ... ۶۵۵

۱۴-۳. مطالعه موردی: طبقه‌بندی با k نزدیک‌ترین همسایگی و دیتاست Digits، بخش ۲ ..... ۵۲۱

۱۴-۳-۱. معیارهای دقت مدل ..... ۵۲۲

۱۴-۳-۲. اعتبارسنجی k لایه‌ای متقابل ..... ۵۲۷

۱۴-۳-۳. اعتبارسنجی k لایه‌ای متقابل ..... ۵۲۸

۱۴-۳-۴. تنظیم هایپر پارامتر ..... ۵۲۹

۱۴-۴. مطالعه موردی: سری زمانی و رگرسیون خطی ساده ..... ۵۳۰

۱۴-۵. مطالعه موردی: رگرسیون خطی چندگانه با دیتاست California Housing ..... ۵۳۶

۱۴-۵-۱. بارگذاری دیتاست ..... ۵۳۶

۱۴-۵-۲. بررسی داده با pandas ..... ۵۳۹

۱۴-۵-۳. نمودار ویژگی‌ها ..... ۵۴۱

۱۴-۵-۴. تقسیم داده برای آموزش و تست ..... ۵۴۶

۱۴-۵-۵. آموزش مدل ..... ۵۴۷

۱۴-۵-۶. تست مدل ..... ۵۴۸

۱۴-۵-۷. نمودار قیمت‌های مورد انتظار در قیاس با قیمت‌های پیش‌بینی شده ..... ۵۴۸

۱۴-۵-۸. معیارهای مدل رگرسیون ..... ۵۵۰

۱۴-۵-۹. انتخاب بهترین مدل ..... ۵۵۱

۱۴-۶. مطالعه موردی: یادگیری ماشینی بدون ناظر، قسمت ۱ - کاهش ابعاد ..... ۵۵۲

۱۴-۷. مطالعه موردی: یادگیری ماشینی بدون ناظر، قسمت ۲ - خوشه‌بندی k میانگین ..... ۵۵۷

۱۴-۷-۱. بارگذاری دیتاست Iris ..... ۵۶۰

۱۴-۷-۲. بررسی دیتاست Iris: آمار توصیفی با pandas ..... ۵۶۲

۱۴-۷-۳. نمودار دیتاست با Seaborn pairplot ..... ۵۶۴

۱۴-۷-۴. استفاده از برآورد کننده KMeans ..... ۵۶۸

۱۴-۷-۵. کاهش ابعاد با تحلیل مؤلفه اساسی ..... ۵۷۰

۱۴-۷-۶. انتخاب بهترین برآورد کننده خوشه‌بندی ..... ۵۷۲

۱۴-۸. جمع‌بندی ..... ۵۷۴

## فصل پانزدهم: یادگیری عمیق

۱۵-۱. مقدمه ..... ۵۷۷

۱۵-۱-۱. کاربردهای یادگیری عمیق ..... ۵۸۰

۱۵-۱-۲. دموهای یادگیری عمیق ..... ۵۸۰

۱۵-۱-۳. منابع کراس ..... ۵۸۱

۱۵-۲. دیتاست‌های داخلی کراس ..... ۵۸۲

۱۵-۳. محیط‌های سفارشی آناکوندا ..... ۵۸۳

۱۵-۴. شبکه‌های عصبی ..... ۵۸۴

۱۵-۵. تنسورها ..... ۵۸۷

۱۶-۳-۳. پایگاه داده‌های NoSQL از نوع چند ستونه .....	۶۵۵
۱۶-۳-۴. پایگاه داده‌های NoSQL از نوع گراف. ۶۵۶.	۶۵۶
۱۶-۳-۵. پایگاه داده‌های NewSQL.....	۶۵۷
۱۶-۴. مورد مطالعاتی - یک پایگاه داده MongoDB	
JSON Document .....	۶۵۸
۱۶-۴-۱. ایجاد یک کلاستر اطلس MongoDB ....	۶۵۹
۱۶-۴-۲. توییت‌ها به MongoDB.....	۶۶۰
۱۶-۵. هادوپ .....	۶۷۰
۱۶-۵-۱. مرور هادوپ .....	۶۷۰
۱۶-۵-۲. خلاصه کردن طول کلمات در نمایشنامه رومئو و ژولیت با استفاده از نگاشت کاهش .....	۶۷۴
۱۶-۵-۳. ایجاد یک هادوپ آپاچی در Azure	
HDInsight مایکروسافت .....	۶۷۴
۱۶-۵-۴. استریمینگ هادوپ .....	۶۷۷
۱۶-۵-۵. پیاده‌سازی Mapper .....	۶۷۷
۱۶-۵-۶. پیاده‌سازی Reducer .....	۶۷۸
۱۶-۵-۷. آماده شدن برای اجرای مثال نگاشت کاهش .....	۶۷۹
۱۶-۵-۸. اجرای Job نگاشت کاهش .....	۶۸۰
۱۶-۶. اسپارک .....	۶۸۳
۱۶-۶-۱. مرور اسپارک .....	۶۸۳
۱۶-۶-۲. Docker و Jupyter Docker Stacks .....	۶۸۵
۱۶-۶-۳. شمارش کلمات با اسپارک .....	۶۸۹
۱۶-۶-۴. شمارش کلمات اسپارک در Azure	
مایکروسافت .....	۶۹۳
۱۶-۷. استریمینگ اسپارک: شمارش هشک‌های توییت با استفاده از pyspark-notebook .....	۶۹۶
۱۶-۷-۱. استریمینگ توییت‌ها به یک سوکت .....	۶۹۷
۱۶-۷-۲. خلاصه کردن هشک‌های توییت: مقدمه‌ای بر SQL اسپارک .....	۷۰۱
۱۶-۸. اینترنت اشیا و داشبوردها .....	۷۰۷
۱۶-۸-۱. انتشار و اشتراک .....	۷۰۹
۱۶-۸-۲. رسم گرافیکی نمونه‌های جریان زنده PubNub با داشبورد Freeboard .....	۷۰۹
۱۶-۸-۳. شبیه‌سازی ترموستات متصل به اینترنت در پایتون .....	۷۱۲
۱۶-۸-۴. ایجاد داشبورد با Freeboard.io .....	۷۱۵
۱۶-۸-۵. ایجاد یک اشتراک PubNub پایتون .....	۷۱۷
۱۶-۹. جمع‌بندی .....	۷۲۲



## مقدمه

### در آن تپه‌ها طلا وجود دارد!

ناشناس، اما اغلب به غلط به مارک تو این نسبت داده می‌شود.

به کتاب پایتون برای برنامه نویسان خوش آمدید! در این کتاب به طور عملی جذاب‌ترین و پیشرفته‌ترین فناوری‌های محاسباتی مدرن را یاد می‌گیرید و برنامه‌نویسی در پایتون را می‌آموزید. پایتون یکی از محبوب‌ترین زبان‌های برنامه‌نویسی جهان محسوب شده و در میان سایر زبان‌ها بیشترین رشد را دارد.

توسعه‌دهندگان اغلب به سرعت درمی‌یابند که پایتون را دوست دارند. این توسعه‌دهندگان از قابلیت کد نویسی راحت و قوی آن، خوانایی، خلاصه بودن و قابلیت تعامل پایتون قدردانی می‌کنند. این توسعه‌دهندگان دنیای توسعه نرم‌افزار متن‌باز<sup>۱</sup> را دوست دارند به دلیل آن که متن‌باز به عنوان یک پایگاه نرم‌افزاری با قابلیت استفاده مجدد در طیف وسیعی از حوزه‌ها به کار می‌رود و به سرعت در حال رشد است. برای چندین دهه، برخی از روندهای سیطره داشته است. سخت‌افزار کامپیوتر به سرعت سریع‌تر، ارزان‌تر و کوچک‌تر شده است. پهنای باند اینترنت به سرعت بزرگ‌تر و ارزان‌تر شده است. نرم‌افزارهای با کیفیت مطلوب از طریق جنبش منبع باز در طیف وسیعی در دسترس هستند و اساساً رایگان یا تقریباً رایگان قابل‌استیاع می‌باشند. به زودی، اینترنت اشیا (IoT) ده‌ها میلیارد دستگاه از هر نوعی که قابل تصور باشد را به هم متصل می‌کند. تمامی این تغییرات حجم عظیمی از داده‌ها را با سرعت و در مقیاس خیلی زیاد تولید می‌کنند. امروزه در دنیای محاسبات، جدیدترین نوآوری‌ها در ارتباط با داده است، برای مثال همه‌چیز در مورد داده‌ها - علوم داده، تجزیه و تحلیل داده‌ها، داده کلان داده، پایگاه داده‌های رابطه‌ای (SQL) و پایگاه‌های داده NoSQL و NewSQL مثال‌های از این دست محسوب می‌شوند. در این کتاب هریک از این مباحث را به همراه یک روش نوآورانه در برنامه‌نویسی پایتون شرح خواهیم داد.

### مشاغل امروز به مهارت‌های علوم داده نیاز دارند

در سال ۲۰۱۱، موسسه جهانی مک کینزی گزارشی خود را با عنوان کلان داده: مرز بعدی برای نوآوری، رقابت و بهره‌وری منتشر کرد. در این گزارش گفتند: ایالات متحده به تنهایی با کمبود ۱۴۰،۰۰۰ تا ۱۹۰،۰۰۰ نفر با مهارت‌های تحلیل عمیق و همچنین یک و نیم میلیون مدیر و تحلیلگر برای تجزیه و تحلیل داده‌های بزرگ و تصمیم‌گیری بر اساس یافته‌های آن‌ها روبرو است. این واقعیت کماکان به قوت خود باقی است. گزارش نیروی کار LinkedIn در آگوست ۲۰۱۸ می‌گوید کمبود ایالات متحده در مورد متخصصین علوم داده بالغ بر ۱۵۰،۰۰۰ است. گزارش سال ۲۰۱۷ IBM، Burning Glass Technologies و BusinessHigher Education Forum، عنوان می‌کند تا سال ۲۰۲۰ در ایالات متحده صدها هزار شغل جدید ایجاد خواهد شد که به مهارت‌های علوم داده نیاز دارند.

[https://www.mckinsey.com/~media/McKinsey/Business%20Functions/McKinsey%20Digital/Our%20Insights/Big%20data%20The%20next%20frontier%20for%20innovation/MGI\\_big\\_data\\_exec\\_summary.pdf](https://www.mckinsey.com/~media/McKinsey/Business%20Functions/McKinsey%20Digital/Our%20Insights/Big%20data%20The%20next%20frontier%20for%20innovation/MGI_big_data_exec_summary.pdf)  
<https://economicgraph.linkedin.com/resources/linkedin-workforce-report-august-2018>  
[https://www.burning-glass.com/wp-content/uploads/The\\_Quant\\_Crunch.pdf](https://www.burning-glass.com/wp-content/uploads/The_Quant_Crunch.pdf)

### معماری ماژولار

معماری ماژولار کتاب (لطفاً به نمودار مطالب داخل جلد کتاب مراجعه کنید) به ما کمک می‌کند تا نیازهای متنوع مخاطبان حرفه‌ای مختلف را برآورده کنیم.

فصل ۰ - ۱ برنامه‌نویسی پایتون را مورد بررسی قرار می‌دهد. هر یک از این فصل‌ها شامل یک بخش مختصر با نام مقدمه‌ای بر علوم داده هستند و مباحثی را مطرح می‌کنند. برای مثال معرفی هوش مصنوعی، آمار توصیفی پایه، سنجش گرایش و پراکندگی مرکزی، شبیه‌سازی، رسم نمودار استاتیک و پویا، کار با فایل‌های CSV، Pandas برای بررسی داده‌ها و آماده‌سازی داده‌ها، سری‌های زمانی و رگرسیون خطی ساده مثال‌هایی از این دست محسوب می‌شوند. این مباحث به شما کمک می‌کنند تا برای مطالعات موردی در زمینه علوم داده، هوش مصنوعی، کلان داده در فصل‌های ۶ تا ۱۱ آماده شوید، در این فصول فرصت‌هایی خواهید داشت تا از دیتاست‌های دنیای واقعی در این موارد مطالعاتی کامل استفاده کنید.

<sup>۱</sup> Open Source

پس از پرداختن به فصل‌های ۱ تا ۵ پایتون و چند بخش اصلی از فصل‌های ۶ تا ۷، می‌توانید بخش‌های قابل توجهی از مطالعات موردی را در فصل‌های ۱۱ - ۶ را بررسی کنید. بخش وابستگی فصل‌ها این مقدمه به مدرسین کمک خواهد کرد تا دوره‌های تدریس خود را در بر اساس معماری منحصر به فرد این کتاب برنامه‌ریزی کنند.

در فصل‌های ۶ - ۱۱ مثال‌های جالب، قدرتمند و جدید زیادی وجود دارد. این مثال‌های پیاده‌سازی عملی مورد‌های مطالعاتی در درباره موضوعاتی از قبیل پردازش زبان طبیعی، داده‌کاوی تویتر، محاسبات شناختی با آی بی آم واتسون، یادگیری ماشین تحت نظارت با طبقه‌بندی و رگرسیون، یادگیری ماشین بدون نظارت با خوشه‌بندی، یادگیری عمیق با شبکه‌های عصبی کانولوشن، یادگیری عمیق با شبکه‌های عصبی بازگشتی، کلان داده با پایگاه داده‌های هادوپ، اسپارک و NoSQL، اینترنت اشیا و موارد دیگر را انجام می‌دهند. در این مسیر، دانش گسترده‌ای در مورد اصطلاحات و مفاهیم علوم داده کسب خواهید کرد، از تعاریف مختصر گرفته تا استفاده از این مفاهیم در برنامه‌های کوچک، متوسط و بزرگ. مرور دقیق فهرست مطالب و فهرست کتاب به شما کمک می‌کند تا درک درستی از مباحث مطرح شده به دست آورید.

### ویژگی‌های کلیدی

**KIS (آن را ساده نگاه دارید)، KIS (آن را کوچک نگاه دارید)، KIT (آن را به روز نگاه دارید)**

❖ **آن را ساده نگاه دارید** - در هر قسمت از کتاب، برای سادگی و وضوح تلاش می‌کنیم. به عنوان مثال، هنگامی که پردازش زبان طبیعی را ارائه می‌دهیم، به جای استفاده از NLTK پیچیده‌تر، از کتابخانه TextBlob ساده و شهودی استفاده می‌کنیم. در ارائه یادگیری عمیق، Keras را به TensorFlow ترجیح می‌دهیم. به طور کلی، وقتی می‌توان از چندین کتابخانه برای انجام کارهای مشابه استفاده کرد، از ساده‌ترین آن استفاده می‌کنیم.

❖ **آن را کوچک نگاه دارید** - بیشتر ۵۳۸ مثال این کتاب، مثال‌های کوچک هستند و اغلب هم فقط در حد چند خط کد بوده و از محیط تعاملی IPython بازخورد فوری دریافت می‌کنند. علاوه بر این در حدود ۴۰ اسکرپت بزرگ‌تر در مطالعات موردی عمیق گنجانده‌ایم.

❖ **آن را به روز نگاه دارید** - تعداد زیادی از کتاب‌های جدید برنامه‌نویسی پایتون و علوم داده را خوانده‌ایم. حدود ۱۵۰۰۰ مقاله، مقاله پژوهشی، گزارش‌های رسمی، فیلم‌ها، پست‌های وبلاگ، پست‌های انجمن و مستندات را خواندیم یا تماشا کردیم. این رویکرد این امکان را به ما داد تا پایتون، علوم کامپیوتر، علوم داده، هوش مصنوعی، کلان داده و انجمن‌های مبتنی بر ابر را عمیقاً درک کنیم.

### بازخورد فوری: کاوش، بررسی و آزمایش با IPython

❖ راه ایدئال برای یادگیری از این کتاب خواندن آن و اجرای مثال‌های کد به‌طور موازی است. در طول کتاب، از مفسر IPython استفاده می‌کنیم. IPython یک محیط کاربرپسند و تعاملی با بازخوردی سریع را برای کاوش، بررسی و آزمایش با پایتون و کتابخانه‌های گسترده آن فراهم می‌کند.

❖ بیشتر کدها در جلسات کوچک و تعاملی IPython ارائه شده‌اند. برای هر اسنپتی از کد که می‌نویسید، IPython بلافاصله آن را می‌خواند، ارزیابی می‌کند و نتایج را چاپ می‌کند. این بازخورد فوری توجه شما را حفظ می‌کند، یادگیری را افزایش می‌دهد، نمونه‌سازی سریع را تسهیل می‌کند و روند توسعه نرم‌افزار را تسریع می‌کند.

❖ کتاب‌های ما همیشه بر روی رویکرد کد آماده تأکید دارند و بر برنامه‌های کامل که ورودی و خروجی زنده کار می‌کنند تمرکز دارند. IPython magic ابزاری است که با وارد کردن هر سطر از اسنپت آن را به کدی آماده تبدیل می‌کند. این رویکرد موجب افزایش کیفیت یادگیری و تشویق به آزمایش می‌شود.

### مبانی برنامه‌نویسی پایتون

- ❖ بررسی کامل و غنی پایتون، اولین و مهم‌ترین موضوع محسوب می‌شود.
- ❖ مدل‌های برنامه‌نویسی پایتون برنامه‌نویسی رویه‌ای، برنامه‌نویسی تابعی و برنامه‌نویسی شیء‌گرا را شرح می‌دهیم.
- ❖ با تأکید بر اصطلاحات مدرن از بهترین روش‌ها استفاده می‌کنیم.
- ❖ در کل کتاب از برنامه‌نویسی به سبک عملکردی استفاده می‌شود. یک جدول در فصل ۴ بیشتر قابلیت‌های اصلی برنامه‌نویسی سبک کاربردی پایتون و فصل‌هایی را که در ابتدا بیشتر آن‌ها را بررسی می‌کنیم، فهرست می‌کند.

## ۵۳۸ نمونه کد

❖ شما با ۵۳۸ مثال دنیای واقعی از اسنپت‌های جداگانه گرفته تا مباحث مهم علوم کامپیوتر، علوم داده، هوش مصنوعی و مطالعات موردی کلان داده، مقدمه‌ای جذاب، چالش‌برانگیز و سرگرم‌کننده برای پایتون کار خواهید کرد.

❖ شما با هوش مصنوعی، کلان داده و فناوری‌های ابری مانند پردازش زبان طبیعی، داده‌کاوی تویتر، یادگیری ماشینی، یادگیری عمیق، هادوپ، کاهش نگاشت، اسپارک، واتسون IBM، کتابخانه‌های کلیدی علوم داده (NumPy، pandas، SciPy، NLTK)، (TextBlob، spacy، Textastic، Tweepy، Scikitlearn، Keras)، کتابخانه‌های کلیدی رسم نمودار (matplotlib، Folium، Seaborn) و موارد دیگر کارهای مهمی انجام خواهید داد.

### اجتناب از ریاضیات سنگین به نفع توضیحات انگلیسی

❖ ما فقط از مفهوم و نتیجه ریاضیات استفاده می‌کنیم و در مثال‌های خود به کار می‌بندیم. این کار را با استفاده از کتابخانه‌هایی مانند Pandas، SciPy، NumPy، statistics و بسیاری از کتابخانه‌های دیگر انجام می‌دهیم؛ این کتابخانه‌ها پیچیدگی ریاضی را پنهان می‌کنند. برای شما ساده است که بسیاری از مزایای تکنیک‌های ریاضی مانند رگرسیون خطی را بدون نیاز به دانستن ریاضیات پشت‌صحنه آن‌ها استفاده کنید. در مثال‌های یادگیری ماشینی و یادگیری عمیق، بر ایجاد اشیایی تمرکز می‌کنیم که در پشت‌صحنه محاسبات ریاضی را برای شما انجام می‌دهند.

### رسم نمودار

❖ ۶۷ مورد نمودارهای گرافیکی ایستا، پویا، انیمیشن و تعاملی (نمودارها، گراف‌ها، تصاویر، انیمیشن‌ها و غیره) در درک مفاهیم به شما کمک می‌کنند.

❖ به‌جای استفاده از برنامه‌نویسی گرافیکی سطح پایین، روی بصری سازی و گرافیک سطح بالای تولیدشده توسط matplotlib، pandas، Seaborn و Folium (برای نقشه‌های تعاملی) تمرکز می‌کنیم.

❖ از نمودار به‌عنوان ابزاری آموزشی استفاده کنید. به‌عنوان مثال، قانون اعداد بزرگ را به‌صورت نمودار در شبیه‌سازی پرتاب یک تاس نشان داده‌ایم. با افزایش تعداد پرتاب‌ها مشاهده خواهید کرد درصد نسبی وقوع تکرار هر وجه، به تدریج به ۱۶٪ (۱/۶) نزدیک می‌شود و ارتفاع میله‌های نشان‌دهنده درصد هر عدد تاس، در نمودار گرافیکی برابر می‌شوند.

❖ گرافیک و رسم نمودار در داده‌های بزرگ برای اکتشاف داده‌ها و انتقال نتایج تحقیقات قابل تکرار که در آن تعداد داده‌ها می‌تواند به میلیون‌ها، میلیاردها یا بیشتر برسند، بسیار مهم است. یک جمله رایج می‌گوید که یک عکس به اندازه هزار کلمه ارزش دارد - در داده‌های بزرگ، یک نمودار می‌تواند میلیاردها، تریلیون‌ها یا حتی بیشتر از دیگر موارد در یک پایگاه داده ارزش داشته باشد. یک نمودار به شما امکان می‌دهد تا چشم‌انداز وسیعی از اطلاعات خود به دست آورید و آن‌ها را بهتر بشناسید. آمار توصیفی کمک می‌کند اما می‌تواند گمراه‌کننده باشد. به‌عنوان مثال، چهارتایی آنسکوم از طریق نمودار نشان می‌دهد که دیتاست‌های می‌توانند به‌طور قابل توجهی متفاوت باشند ولی آمار توصیفی تقریباً یکسانی داشته باشند.

❖ [https://en.wikipedia.org/wiki/A\\_picture\\_is\\_worth\\_a\\_thousand\\_words](https://en.wikipedia.org/wiki/A_picture_is_worth_a_thousand_words).

❖ [https://en.wikipedia.org/wiki/Anscombe%27s\\_quartet](https://en.wikipedia.org/wiki/Anscombe%27s_quartet)

❖ کدهای رسم نمودار و انیمیشن را نشان می‌دهیم تا بتوانید کد خود را پیاده‌سازی کنید. همچنین انیمیشن‌ها را در فایل‌های سورس کد و به‌عنوان نوت‌بوک‌های Jupyter ارائه می‌دهیم، بنابراین می‌توانید به‌راحتی کد و پارامترهای انیمیشن را سفارشی کنید، انیمیشن‌ها را دوباره اجرا کنید و تأثیر تغییرات را ببینید.

❖ بخش مقدمه‌ای برای علوم داده و مطالعات موردی در فصول ۶ تا ۱۱، تجربیات ارزشمندی در زمینه داده ارائه می‌دهند.

❖ شما با بسیاری از دیتاست‌های واقعی و منابع داده کار خواهید کرد. مجموعه گسترده‌ای از دیتاست‌های باز و رایگان به‌صورت آنلاین برای آزمایش وجود دارد. برخی از سایت‌هایی که به آن‌ها مراجعه می‌کنیم صدها یا هزاران دیتاست را لیست می‌کنند.

❖ بسیاری از کتابخانه‌هایی که آن‌ها استفاده می‌کنید همراه با دیتاست‌های رایج برای آزمایش هستند.

❖ مراحل لازم برای به دست آوردن داده‌ها و آماده‌سازی آن‌ها را برای تجزیه و تحلیل یاد خواهید گرفت، با استفاده از بسیاری از تکنیک‌ها داده‌ها را تجزیه و تحلیل خواهید کرد، مدل‌های خود را تنظیم می‌کنید و نتایج خود به‌طور مؤثر انتقال می‌دهید و تمامی این - کارها را به‌طور خاص از طریق بصری سازی و رسم نمودار انجام خواهید داد.

## گیت‌هاب<sup>۲</sup>

❖ گیت‌هاب مکانی عالی برای یافتن کد و گنجاندن آن‌ها در پروژه‌هایتان (و اشتراک گذاشتن کدتان در انجمن منبع باز) محسوب می‌شود. همچنین گیت‌هاب یک عنصر مهم در مجموعه نرم‌افزار توسعه‌دهندگان با ابزارهای کنترل ورژن محسوب می‌شود و به تیم توسعه‌دهندگان کمک می‌کند تا پروژه‌های منبع باز (و خصوصی) را مدیریت کنند.

❖ از طیف وسیعی از کتابخانه‌های رایگان و منبع باز علوم داده، پایتون و خدمات رایگان، رایگان آزمایشی و رایگان پایه نرم‌افزارها و سرویس‌های ابری استفاده خواهید کرد.

## محاسبات عملی در ابر

❖ تجزیه و تحلیل کلان داده‌ها در فضای ابری انجام می‌گیرد و در چنین فضایی، سخت‌افزار و نرم‌افزار مورد نیاز برنامه‌های به‌طور پویا و راحت تغییر می‌کند. شما با سرویس‌های مختلف مبتنی بر ابر (برخی مستقیم و برخی غیرمستقیم)، از جمله توئیتر، مترجم گوگل، واتسون IBM، Azure مایکروسافت، OpenMapQuest، geopy، Dweet.io و PubNub کار خواهید کرد.

❖ شما را تشویق می‌کنیم از سرویس‌های رایگان آزمایشی رایگان، یا رایگان پایه استفاده کنید. شخصاً سرویس‌هایی را ترجیح می‌دهیم که به کارت اعتباری احتیاج ندارند زیرا نمی‌خواهیم به‌طور تصادفی هزینه زیادی را متحمل شوید. اگر تصمیم دارید از سرویسی که به کارت اعتباری نیاز دارد استفاده کنید، اطمینان حاصل کنید خدماتی که به‌صورت رایگان استفاده می‌کنید به‌طور خودکار به خدمات مشمول هزینه تبدیل نخواهد شد.

## پایگاه داده، کلان داده و زیرساخت‌های کلان داده

❖ طبق گفته IBM (نوامبر ۲۰۱۶)، ۹۰ درصد از داده‌های جهان در دو سال گذشته ایجاد شده است. شواهد نشان می‌دهند که سرعت ایجاد داده‌ها به‌سرعت در حال افزایش است.

❖ <https://public.dhe.ibm.com/common/ssi/ecm/wr/en/wr112345usen/watson-customer-engagement--watson-marketing-wr-other-papers-and-reports-r112345usen-20170719.pdf>

❖ طبق مقاله تجزیه و تحلیل هفتگی در مارس ۲۰۱۶، طی ۵ سال بیش از ۵۰ میلیارد دستگاه به اینترنت متصل می‌شوند و تا سال ۲۰۲۰ هر ثانیه ۱,۷ مگابایت داده جدید برای هر فرد روی کره زمین تولید خواهیم کرد!

<https://analyticsweek.com/content/big-data-facts/>

❖ SQLite را به‌عنوان پایگاه داده رابطه‌ای همراه با SQL مورد بررسی قرار خواهیم داد.

❖ پایگاه‌های داده از زیرساخت‌های مهم داده‌های بزرگ برای ذخیره و کار با مقادیر گسترده‌ای از داده‌های هستند که پردازش می‌کنید. پایگاه‌های داده رابطه‌ای داده‌های دارای ساختار را پردازش می‌کنند و برای کار با داده‌های بدون ساختار و یا نیمه ساختار مناسب نیستند، بنابراین، با افزایش روزافزون کلان داده، پایگاه داده‌های NoSQL و NewSQL ایجاد شدند تا چنین داده‌هایی را به‌طور کارآمدی مدیریت کنند. یک مرور کلی در خصوص NoSQL و NewSQL و یک مطالعه موردی با پایگاه داده اسناد MongoDB ارائه خواهیم داد. MongoDB یکی از محبوب‌ترین پایگاه داده NoSQL محسوب می‌شود.

❖ در بخش ۱۶ با عنوان "کلان داده، هادوپ، اسپارک، NoSQL و IoT (اینترنت اشیا)" زیرساخت سخت‌افزاری و نرم‌افزاری کلان داده را مورد بررسی قرار خواهیم داد.

## مطالعات موردی هوش مصنوعی

❖ در مطالعه موردی فصل‌های ۵ تا ۱۱، مباحث هوش مصنوعی، از جمله پردازش زبان طبیعی، داده‌کاوی توئیتر برای انجام تجزیه و تحلیل احساسات، محاسبات شناختی با واتسون IBM، یادگیری ماشین با ناظر، یادگیری ماشین بدون ناظر و یادگیری عمیق را

ارائه می‌دهیم. فصل ۱۶ زیرساخت‌های سخت‌افزاری و نرم‌افزاری داده‌های بزرگ را ارائه می‌دهد، این زیرساخت‌ها دانشمندان کامپیوتر و دانشمندان علوم داده را قادر می‌سازد تا راه‌حل‌های مبتنی بر هوش مصنوعی ارائه دهند.

### **کالکشن‌های داخلی: لیست‌ها، چندتایی‌ها، مجموعه‌ها، دیکشنری‌ها**

❖ امروزه دلیل کمی برای بیشتر توسعه‌دهندگان نرم‌افزار وجود دارد تا ساختمان داده‌های سفارشی را ایجاد کنند. دو فصل این کتاب ساختمان داده‌های داخلی پایتون مانند - چندتایی‌ها، مجموعه‌ها، دیکشنری‌ها را به تفصیل موردبررسی قرار می‌دهد و با آن‌ها می‌توان بیشتر کارهای مربوط به ساختارهای داده را انجام داد.

### **برنامه‌نویسی آرایه‌ها با آرایه‌های NumPy و series / دیتافریم‌ها در Pandas**

❖ علاوه بر این، روی سه ساختار کلیدی داده از کتابخانه منبع باز - آرایه‌های NumPy، سری pandas و دیتافریم‌های pandas تمرکز می‌کنیم. این سه ساختار در علوم داده، علوم کامپیوتر، هوش مصنوعی و کلان داده بسیار استفاده می‌شود. NumPy در قیاس با لیست‌های داخلی پایتون از کارایی و سرعتی دو برابر برخوردار است.

❖ در فصل ۷ آرایه‌های NumPy را به‌طور کامل بررسی خواهیم کرد. بسیاری از کتابخانه‌ها مانند Pandas بر اساس NumPy ساخته شده‌اند. در بخش مقدمه‌ای برای علوم داده در فصل ۷ - ۹ سری pandas و دیتافریم‌های pandas را معرفی می‌کنیم و در تمامی فصول باقی‌مانده از آرایه‌های NumPy و این دو کتابخانه استفاده خواهیم کرد.

### **پردازش فایل و سریال سازی**

❖ فصل ۹ پردازش فایل متنی را ارائه می‌دهد و سپس نحوه سریال سازی اشیا را با استفاده از فرمت محبوب JSON (JavaScript Object Notation) نشان می‌دهد. JSON به کرات در فصول علوم داده مورد استفاده قرار می‌گیرد.

❖ بسیاری از کتابخانه‌های علوم داده، قابلیت پردازش فایل را به‌طور داخلی برای بارگذاری دیتاست‌ها در برنامه‌های پایتون فراهم می‌کنند. علاوه بر فایل‌های متنی ساده، با استفاده از ماژول csv کتابخانه استاندارد پایتون و قابلیت‌های کتابخانه علوم داده pandas، پروتج‌ها را در قالب رایج CSV (comma-separated values) پردازش می‌کنیم.

### **برنامه‌نویسی مبتنی بر شیء**

❖ تاکید می‌کنیم که از تعداد زیادی کلاس باارزش استفاده کنید که انجمن منبع باز پایتون در کتابخانه‌های کلاس استاندارد صنعت بسته‌بندی کرده است. باید بدانید که چه کتابخانه‌هایی در آن جا وجود دارد، مواردی که برای برنامه‌های خود نیاز دارید انتخاب کنید، اشیا را از کلاس‌های موجود (معمولاً در یک یا دو خط کد) ایجاد کنید و مشکلاتان را به راحتی مرتفع کنید. این سبک برنامه‌نویسی که مبتنی بر شیء نام دارد و این امکان را به شما می‌دهد تا به سرعت و به‌طور مختصر برنامه‌های مطلوبی بسازید و این ویژگی قسمت عمده‌ای از دلایل جذابیت‌های پایتون محسوب می‌شود.

❖ با این رویکرد، می‌توانید با استفاده از یادگیری ماشینی، یادگیری عمیق و سایر فناوری‌های هوش مصنوعی طیف گسترده‌ای از مسئله‌های جذاب همانند چالش‌های محاسبات شناختی مانند تشخیص گفتار و بینایی ماشینی را به سرعت حل کنید.

### **برنامه‌نویسی شیء‌گرا**

❖ توسعه کلاس‌های سفارشی، همراه با ارث‌بری، چندشکلی و duck typing یکی از مهارت‌های اساسی در برنامه‌نویسی شیء‌گرا محسوب می‌شوند. در فصل ۱۰ این مباحث را بررسی می‌کنیم.

❖ فصل ۱۰ در مورد تست واحد با doctest و شبیه‌سازی برزدن کارت‌ها و توزیع آن‌ها شرح می‌دهد.

❖ در فصل‌های ۶ تا ۱۱ فقط به چند تعریف کلاس سفارشی ساده نیاز دارند. در پایتون، احتمالاً از رویکرد برنامه‌نویسی مبتنی بر شیء بیشتر از برنامه‌نویسی شیء‌گرا استفاده خواهید کرد.

### **تجدید پذیری**

❖ در علوم به‌طور کلی و به‌طور خاص در علوم داده، نیاز به بازتولید نتایج آزمایشات، مطالعات و برقراری ارتباط مؤثر آن نتایج است. نوت‌بوک‌های Jupyter ابزار ترجیحی برای انجام این کار محسوب می‌شوند.

❖ در کل کتاب در مورد تجدید پذیری در زمینه تکنیک‌های برنامه‌نویسی و نرم‌افزار مانند نوت‌بوک‌های Jupyter و docker صحبت می‌کنیم.

## کارایی

❖ برای مقایسه کارایی رویکردهای مختلف برای انجام کارهای یکسان، از چندین نمونه ابزار `timeit` استفاده می‌کنیم. سایر بحث‌های مربوط به کارایی شامل مولد عبارات، آرایه‌های `Numpy` در قیاس با لیست‌های پایتون، کارایی مدل‌های یادگیری ماشینی و یادگیری عمیق و کارایی هادوپ و اسپارک در محاسبات توزیع شده هستند.

## کلان داده و توزی

❖ در این کتاب به جای نوشتن کد موازی‌سازی خودتان، اجازه می‌دهید کتابخانه‌هایی مانند اسپارک که روی `TensorFlow` کار می‌کنند و ابزارهای کلان داده مانند هادوپ و اسپارک عملیات موازی‌سازی را برای شما انجام دهند. در این دوره کلان داده/هوش مصنوعی، پردازش داده‌های زیاد این برنامه‌ها به پردازش کارآمدی نیاز دارند، این پردازش‌ها به نوبه خود به موازی‌سازی قوی ارائه شده توسط پردازنده‌های چند هسته‌ای، واحدهای پردازش گرافیکی (`gpu`)، واحدهای پردازش `TPUs` (تانسور) و کلاسترهای عظیم کامپیوتری در ابر نیاز دارند. برخی از کارهای کلان داده می‌تواند هزاران پردازنده را داشته باشد که به صورت موازی کار می‌کنند تا مقادیر انبوهی از داده را با سرعت تجزیه و تحلیل کنند.

## وابستگی فصل‌ها

❖ اگر یک مدرس هستید که برنامه درسی خود را برای یک دوره آموزش حرفه‌ای برنامه‌ریزی می‌کنید یا یک توسعه‌دهنده هستید که تصمیم می‌گیرید که کدام فصل‌ها را بخواند، این بخش به شما کمک می‌کند تا بهترین تصمیم را اتخاذ کنید. لطفاً صفحه رنگی محتویات کتاب را مطالعه کنید، این فهرست به سرعت شما را با معماری منحصر به فرد کتاب آشنا می‌سازد. آموزش یا خواندن فصل‌ها به ترتیب آسان‌ترین روش محسوب می‌شود. با این حال، بیشتر مطالب در بخش مقدمه‌ای برای علوم داده در انتهای فصل‌های ۱ تا ۱۰ و مطالعات موردی در فصل‌های ۱۱ - ۱۶ فقط به فصل‌های ۱ تا ۵ و قسمت‌های کمی از فصل‌های ۶ تا ۱۰ نیاز دارند که در ادامه بحث شده است.

## قسمت ۱: شروع سریع مبانی پایتون

توصیه می‌کنیم همه فصل‌ها را به ترتیب بخوانید:

❖ **فصل ۱، مقدمه‌ای بر کامپیوترها و پایتون**، مفاهیمی را معرفی می‌کند که زمینه‌ای برای برنامه‌نویسی پایتون در فصل‌های ۲ تا ۱۰ و کلان داده، هوش مصنوعی و مطالعات مبتنی بر ابر در فصول ۱۱ تا ۱۶ فراهم می‌کند. این فصل همچنین شامل تست‌های موردی از مفسر `IPython` و نوت‌بوک‌های `Jupyter` است.

❖ **فصل ۲، مقدمه‌ای بر برنامه‌نویسی پایتون**، اصول برنامه‌نویسی پایتون را با مثال‌هایی از کد شرح می‌دهد، این مثال‌ها ویژگی‌های اصلی این زبان را نشان می‌دهند.

❖ **فصل ۳، دستورهای کنترلی**، دستورهای کنترلی پایتون را شرح می‌دهد و مبانی پردازش لیست را معرفی می‌کند.

❖ **فصل ۴، توابع**، توابع سفارشی را معرفی می‌کند، تکنیک‌های شبیه‌سازی را با مولد اعداد تصادفی شرح می‌دهد و مبانی چندتایی را معرفی می‌کند.

❖ **فصل ۵، دنباله‌ها: لیست‌ها و چندتایی‌ها**، کالکشن‌های داخلی لیست و چندتایی‌ها در پایتون را جزئیات بیشتری شرح می‌دهد و معرفی برنامه‌نویسی به سبک تابعی را آغاز می‌کند.

## قسمت ۲: ساختمان داده‌های پایتون، رشته‌ها و فایل‌ها

در ادامه به طور خلاصه وابستگی‌های بین فصل‌های پایتون در فصل‌های ۶ تا ۹ را نشان می‌دهد و فرض بر آن است که شما فصل‌های ۱ تا ۵ را خوانده‌اید.

❖ **فصل ۶، دیکشنری‌ها و مجموعه‌ها** - بخش مقدمه‌ای برای علوم داده در این فصل به مطالب این فصل بستگی ندارد.

❖ **فصل ۷، برنامه‌نویسی آرایه‌گر با Numpy** - بخش مقدمه‌ای برای علوم داده به دیکشنری‌ها (فصل ۶) و آرایه‌ها (فصل ۷) نیاز دارد.

❖ **فصل ۸، رشته‌ها: نگاهی عمیق‌تر** - به رشته‌های خام و عبارات منظم (بخش ۸-۱۱ تا ۸-۱۲) و ویژگی‌های سری `pandas` و دیتا فریم از ۷-۱۴ با نام معرفی علوم داده نیاز دارد.

❖ **فصل ۹، فایل‌ها و استثناها** – برای سریال سازی JSON، درک اصول دیکشنری مفید است (بخش ۲-۶). همچنین، بخش مقدمه‌ای بر علوم داده به تابع open، دستور (بخش ۳-۹) with ویژگی‌های دیتا فریم Pandas در قسمت مقدمه‌ای بر علوم داده از بخش ۷، ۱۴ نیاز دارد.

### قسمت ۳: مباحث پیشرفته در پایتون

در ادامه به‌طور خلاصه وابستگی‌ها بین فصل‌های پایتون برای فصل ۱۰ را نشان می‌دهد و فرض بر آن است که شما فصل‌های ۱ تا ۵ را خوانده‌اید.

❖ **فصل ۱۰، برنامه‌نویسی شیء‌گرا** – به بخش مقدمه‌ای برای علوم داده ویژگی‌های دیتا فریم Pandas در بخش ۷-۱۴ در نیاز دارد. مدرسانی که می‌خواهند فقط کلاس‌ها و اشیاء را تدریس کنند می‌توانند بخش‌های ۱-۱۰ تا ۶-۱۰ را آموزش دهند. مدرسانی که می‌خواهند مباحث پیشرفته‌تری مانند وراثت، چندشکلی و duck typing را تدریس کنند، می‌توانند از بخش‌های ۷-۱۰ و ۹-۱۰ استفاده کنند. بخش‌های ۱۰-۱۰ و ۱۵-۱۰ مباحث پیشرفته افزون‌تری ارائه می‌دهند.

### قسمت ۴: مطالعات موردی هوش مصنوعی، ابر و کلان داده

در ادامه به‌طور خلاصه وابستگی‌های بین فصل‌های پایتون در فصل‌های ۱۱ تا ۱۶ را نشان می‌دهد و فرض بر آن است که شما فصل‌های ۱ تا ۵ را خوانده‌اید. علاوه بر این قسمت عمده‌ای از فصل‌های ۱۱ تا ۱۶ به مبانی دیکشنری‌ها در بخش ۲، ۶ نیاز دارند.

❖ **فصل ۱۱، پردازش زبان طبیعی (NLP)**، از ویژگی‌های دیتا فریم pandas از مبحث مقدمه‌ای بر علوم داده در بخش ۷-۱۴ استفاده می‌کند.

❖ **فصل ۱۲، داده‌کاوی توییت‌ها**، از ویژگی‌های دیتا فریم pandas، متد join رشته (بخش ۹-۸) اصول JSON (بخش ۵-۹)، TextBlob (بخش ۲-۱۱) و ابرهای کلمات (بخش ۳-۱۱) استفاده می‌کند. چند مثال در این فصل به تعریف کلاس از طریق وراثت نیاز دارند (فصل ۱۰)

❖ **فصل ۱۳، واتسون IBM و محاسبات شناختی** از توابع درونی open و دستور with (بخش ۳-۹) استفاده می‌کند.

❖ **فصل ۱۴، یادگیری ماشین: طبقه‌بندی، رگرسیون و خوشه‌بندی**، از مبانی آرایه NumPy و متد unique (فصل ۷)، ویژگی‌های دیتا فریم‌های Pandas از مبحث معرفی مقدمه‌ای برای علوم داده در بخش ۷-۱۴ و تابع subplots در matplotlib (بخش ۶-۱۰) استفاده می‌کند.

❖ **فصل ۱۵، یادگیری عمیق**، به اصول آرایه NumPy (فصل ۷)، متد join رشته (بخش ۹-۸)، مفاهیم کلی یادگیری ماشینی فصل ۱۴ و مطالعه موردی فصل ۱۴، طبقه‌بندی با k نزدیک‌ترین همسایگی و دیتاست Digits نیاز دارد.

❖ **فصل ۱۶، کلان داده: هادوپ، اسپارک، NoSQL و IoT**، به متد split رشته (بخش ۷-۲-۶) تابع FuncAnimation در matplotlib از قسمت مقدمه‌ای بر علوم داده در بخش ۴-۶، سری‌ها در pandas و ویژگی‌های دیتا فریم در pandas از قسمت مقدمه‌ای بر علوم داده در بخش ۷-۱۴ متد join رشته (بخش ۹-۸)، ماژول json (بخش ۵-۹)، ایست واژه‌ها NLTK (بخش ۱۳-۲-۱) و برای بخش ۱۲، احراز هویت توییت‌ها، کلاس StreamListener برای استریمینگ توییت‌های و کتابخانه‌های geopy و folium تعدادی از مثال‌ها نیاز به تعریف کلاس از طریق وراثت (فصل ۱۰) نیاز دارد، اما شما می‌توانید بدون خواندن فصل ۱۰ تعاریف کلاس را انجام دهید.

### نوت‌بوک‌های JUPYTER

برای راحتی شما، نمونه‌های کد کتاب را در فایل‌های کد منبع پایتون (.py) برای استفاده با خط فرمان IPython و به‌عنوان فایل‌های نوت‌بوک (.ipynb) Jupyter ارائه می‌دهیم و می‌توانید آن‌ها را در مرورگر وب خودبارگیری کرده و اجرا کنید. نوت‌بوک‌های Jupyter یک پروژه رایگان و منبع باز است که به شما امکان می‌دهد متن، گرافیک، صدا، ویدئو و قابلیت کد نویسی تعاملی را برای ورود، ویرایش، اجرا، اشکال‌زدایی و اصلاح کد به‌سرعت و به‌راحتی در یک مرورگر وب ترکیب کنید. با توجه به **مقاله Jupyter چیست؟**

Jupyter به استانداردی برای تحقیقات علمی و تجزیه و تحلیل داده‌ها تبدیل شده است. Jupyter محاسبات و استدلال‌ها را باهم بسته‌بندی می‌کند و به شما امکان می‌دهد توصیف‌های محاسباتی ایجاد کنید و این مسئله مشکل توزیع نرم‌افزار در حال کار را برای اعضای تیمی و شرکا ساده می‌کند.

❖ <https://www.oreilly.com/ideas/what-is-jupyter>

و طبق تجربه ما، Jupyter یک محیط یادگیری فوق‌العاده و ابزار نمونه‌سازی سریع است. به همین دلیل، به جای استفاده از یک IDE سنتی، مانند Eclipse، Visual Studio، PyCharm یا Spyder از نوت‌بوک‌های Jupyter استفاده می‌کنیم. دانشگاهیان و متخصصان از Jupyter به‌طور گسترده برای اشتراک نتایج تحقیقات استفاده می‌کنند. پشتیبانی از نوت‌بوک‌های Jupyter از طریق سازوکارهای جامعه Open Source سنتی ارائه می‌شود (بعداً در این مقدمه به قسمت "دریافت راهنما برای Jupyter" مراجعه کنید). برای جزئیات نصب نرم‌افزار به بخش قبل از شروع مراجعه کنید، این قسمت این مقدمه را دنبال می‌کند و قسمت ۱،۵ را در مورد اجرای تست مشاهده کنید، این قسمت نحوه اجرای مثال‌های این کتاب را نشان می‌دهد.

❖ <https://jupyter.org/community>.

### همکاری و به اشتراک‌گذاری نتایج

کار کردن در گروه‌ها و به اشتراک گذاشتن نتایج تحقیقات برای توسعه‌دهندگان فعال در تجزیه و تحلیل داده‌ها در صنعت، مشاغل دولتی یا مجامع دانشگاهی و یا توسعه‌دهندگانی که قصد فعالیت در چنین جایگاهی دارند اهمیت ویژه‌ای دارد:

❖ نوت‌بوک‌هایی که ایجاد می‌کنید از کپی کردن فایل‌ها یا از طریق گیت‌هاب به راحتی در میان اعضای تیم به اشتراک گذاشته می‌شوند.

❖ نتایج تحقیق، اعم جمله کد و بینش را می‌توان به عنوان صفحات وب استاتیک و از طریق ابزاری مانند nbviewer (<https://nbviewer.jupyter.org>) و گیت‌هاب به اشتراک گذاشته شود؛ این دو ابزار به‌طور خودکار نوت‌بوک‌ها را به صفحات وب تبدیل می‌کنند.

### تکرارپذیری: یک مورد قوی برای نوت‌بوک‌های Jupyter

در علم داده و به‌طور کلی در علوم، آزمایش‌ها و مطالعات باید قابل تکرار باشند. درباره این موضوع طی این سال‌ها مقاله‌های زیادی نوشته شده است:

❖ علوم کامپیوتر دونالد کنوت در سال ۱۹۹۲ - برنامه‌نویسی ادیبانه<sup>۳</sup>

Knuth, D., "Literate Programming" (PDF), The Computer Journal, British Computer Society, 1992

❖ مقاله تجزیه و تحلیل داده‌های تکرارپذیر زبان با استفاده از برنامه‌نویسی ادیبانه و مستقل از زبان، می‌گوید، lir<sup>۴</sup> (ادیبانه، محاسبات تکرارپذیر) بر اساس ایده برنامه‌نویسی ادیبانه است که توسط دونالد نوت پیشنهاد شده است.

❖ <http://journals.plos.org/plosone/article?d=10.1371/journal.pone.0164023>

اساساً، قابلیت تولید مجدد، محیط کاملی را که برای تولید نتایج استفاده می‌شود، ضبط می‌کند - سخت‌افزار، نرم‌افزار، ارتباطات، الگوریتم‌ها (به‌ویژه کد)، داده‌ها و حاکمیت داده‌ها (مبدأ و ریشه).

### DOCKER

در فصل ۱۶ از docker استفاده خواهیم کرد، docker ابزاری است برای بسته‌بندی نرم‌افزار و تمامی چیزهایی که برای اجرای آسان و راحت نرم‌افزار، قابلیت تولید مجدد و قابلیت حمل در سیستم‌عامل‌ها نیاز می‌باشد را در کانتینرها بسته‌بندی می‌کند. برخی از بسته‌های نرم‌افزاری که در فصل ۱۶ استفاده می‌کنیم نیاز به تنظیم و پیکربندی پیچیده‌ای دارند. برای بسیاری از موارد می‌توانید کانتینرهای رایگان docker موجود را دانلود کنید. این کانتینرها به شما این امکان را می‌دهند با مشکلات پیچیده نصب مواجه نشوید و نرم‌افزار را به صورت محلی روی کامپیوترهای رومیزی یا نوت‌بوک خود اجرا کنید، در حقیقت docker راهی عالی برای کمک به شما در شروع سریع و راحت فن‌آوری‌های جدید محسوب می‌شود.

<sup>۳</sup> Literate Programming

<sup>۴</sup> literate, reproducible computing



علاوه بر این docker به قابلیت تولید مجدد کمک می کند. می توانید کانتینرهای docker سفارشی ایجاد کنید که با نسخه های هر نرم افزار و هر کتابخانه ای که در مطالعه خود استفاده کرده اید، پیکربندی شده باشد. این رویکرد به سایر توسعه دهندگان این امکان را می دهد تا محیطی را که شما استفاده کرده اید از نو بسازند، سپس کار شما را تولید کنند و به شما کمک می کند نتایج خود را بازتولید کنید. در فصل ۱۶، از docker برای دانلود و اجرای کانتینر استفاده خواهید کرد که از قبل برای کد نویسی و اجرای برنامه های کلان داده اسپارک با استفاده از نوت بوک های Jupyter برای شما پیکربندی شده است.

### ویژگی خاص: تجزیه و تحلیل واتسون IBM و محاسبات شناختی

در اوایل تحقیق برای این کتاب، اهمیت رشد روزافزون واتسون را درک کردیم. بسیاری از سرویس های رقیب را مورد بررسی قرار دادیم و به این نتیجه رسیدیم که سیاست واتسون مبنی بر "به هیچ کارت اعتباری نیازی نیست" و خدمات رایگان آن برای خوانندگان ما بهترین گزینه محسوب می شود.

واتسون IBM یک پلت فرم محاسبات شناختی است که در طیف گسترده ای از سناریوهای دنیای واقعی به کار گرفته می شود. سیستم های محاسبات شناختی با استفاده از مقادیر زیادی داده یاد می گیرند و قابلیت های شناخت الگو و تصمیم گیری مغز انسان را شبیه سازی می کنند. یک مثال عملی برای قابلیت های واتسون ارائه خواهیم داد. از Watson Developer Cloud: Python SDK رایگان استفاده می کنیم، این SDK پایتون، API هایی را فراهم می کند و شما را قادر می سازد با سرویس های واتسون از طریق برنامه ارتباط برقرار کنید. استفاده از واتسون سرگرم کننده است و یک بستر عالی برای افزایش خلاقیت شما است. شما می توانید از نسخه دمو و یا API های واتسون برای این موارد استفاده کنید: مکالمه، کشف، مترجم زبان، طبقه بندی کننده زبان طبیعی، درک زبان طبیعی، بینش شخصیت، تبدیل گفتار به متن، تبدیل متن به گفتار، آنالیز صدا و تشخیص بصری.

❖ <http://whatis.techtarget.com/definition/cognitive-computing>.  
❖ [https://en.wikipedia.org/wiki/Cognitive\\_computing](https://en.wikipedia.org/wiki/Cognitive_computing).  
❖ <https://www.forbes.com/sites/bernardmarr/2016/03/23/what-everyone-should-know-about-cognitive-computing>.

### سرویس های رایگان واتسون و یک مورد مطالعاتی جالب

IBM با ارائه خدمات رایگان برای بسیاری از API های خود، یادگیری و آزمایش آن ها را تشویق می کند. در فصل ۱۳، نسخه های دمو بسیاری از سرویس های واتسون را امتحان خواهید کرد. در ادامه از خدمات رایگان متن به گفتار، گفتار به متن و ترجمه واتسون برای پیاده سازی برنامه مترجم سفری استفاده خواهید کرد. سؤالی را به زبان انگلیسی می پرسید، سپس برنامه شما گفتار انگلیسی را به متن انگلیسی تبدیل می کند، متن انگلیسی را به اسپانیایی ترجمه می کند و متن اسپانیایی را به گفتار تبدیل کرده و آن را پخش می کند. در مرحله بعدی، به زبان اسپانیایی پاسخ می دهید (در صورت عدم تسلط به زبان اسپانیایی، یک فایل صوتی ارائه می دهیم که می توانید استفاده کنید). سپس، برنامه به سرعت گفتار را به متن اسپانیایی تبدیل می کند، متن را به انگلیسی ترجمه می کند و پاسخ را به انگلیسی را بیان می کند. **چه جالب!** همیشه آخرین شرایط استفاده از خدمات را در وب سایت آی بی ام بررسی کنید زیرا ممکن است این شرایط و سرویس ها تغییر کنند.

❖ <https://console.bluemix.net/catalog>

### روش آموزش

کتاب پایتون برای برنامه نویسان، شامل مجموعه ای غنی از مثال ها است که از زمینه های مختلف و گسترده ای اقتباس گرفته شده است. با استفاده از دیتاست های دنیای واقعی با مثال های جالب و واقعی کار خواهید کرد. این کتاب بر اصول مهندسی نرم افزار خوب تمرکز دارد و بر وضوح برنامه تأکید می کند.

### استفاده از فونت ها برای تأکید

برای ارجاع راحت تر، اصطلاحات کلیدی و مرجع ایندکس صفحه را برای هر تعریف مشخص در متن پررنگ قرار می دهیم. برای ارجاع به اجزای روی صفحه نمایش از فونت Helvetica پررنگ (به عنوان مثال، منوی File) استفاده کرده و برای کد پایتون از قلم Lucida (به عنوان مثال،  $x = 5$ ) استفاده می کنیم.

### قواعد رنگ بندی

برای افزایش خوانایی، همه کدها را رنگ بندی می کنیم. قراردادهای رنگ آمیزی ما به شرح زیر است:

comments appear in green keywords appear in dark blue

constants and literal values appear in light blue errors appear in red  
all other code appears in black

### ۵۳۸ نمونه کد

۵۳۸ مثال کتاب تقریباً ۴۰۰۰ شامل خط کد می‌باشند. این حجم کد برای کتابی با این اندازه مقدار کمی است و به این دلیل است که زبان پایتون زبانی توصیفی است. همچنین سبک کد نویسی ما یعنی استفاده از کتابخانه‌های کلاس قدرتمند برای بیشتر کارها تا حد ممکن، موجب کاهش حجم کدها شده است.

### ۱۶۰ جدول / نمودار / شکل

در این کتاب جدول‌های فراوان، نمودارهای ایستا، پویا و تعاملی قرار داده‌ایم.

### دانش برنامه‌نویسی

در بحث دانش برنامه‌نویسی، با تلفیق نه دهه تجربه مؤلفین کتاب در زمینه برنامه‌نویسی و تدریس آن به موارد ذیل اشاره می‌کنیم:  
❖ **شیوه‌های خوب برنامه‌نویسی** و اصطلاحات ترجیحی پایتون به شما کمک می‌کند برنامه‌های واضح‌تر، قابل فهم‌تر و قابل نگهداری بیشتری تولید کنید.

❖ **خطاهای برنامه‌نویسی متداول** به شما کمک می‌کند احتمال خطا را کاهش دهید.

❖ **نکات پیشگیری از خطا** همراه با پیشنهادهایی برای مشخص کردن اشکالات و حذف آن‌ها از برنامه‌های شما. بسیاری از این نکات در وهله اول تکنیک‌هایی را شرح می‌دهند تا چنین مشکلاتی در برنامه‌های شما ایجاد نشوند.

❖ **نکاتی درباره کارایی** موقعیت‌هایی را برای اجرای سریع‌تر برنامه‌های شما و یا به حداقل رساندن میزان حافظه مصرفی آن‌ها برجسته می‌کند.

❖ **نکات مهندسی نرم‌افزار**، موضوعات معماری و طراحی را برای ساخت نرم‌افزار مناسب، به‌ویژه برای سیستم‌های بزرگ‌تر برجسته می‌کند

### نرم‌افزار مورد استفاده در این کتاب

نرم‌افزاری که ما استفاده می‌کنیم برای ویندوز، macOS و لینوکس در دسترس بوده و به‌طور رایگان از اینترنت قابل دانلود کردن است. مثال‌های کتاب را با استفاده از توزیع رایگان آنکوندا پایتون نوشته‌ایم. این توزیع پایتون شامل بسیاری از کتابخانه‌های پایتون، رسم نمودار و علوم داده مورد نیاز شما و همچنین مفسر IPython، نوت‌بوک‌های Jupyter و Spyder می‌باشد و در حقیقت یکی از بهترین IDE های پایتون در علوم داده محسوب می‌شود. برای توسعه نرم‌افزار تهیه از نوت‌بوک‌های IPython و Jupyter استفاده می‌کنیم. ادامه این مقدمه در مورد نصب آنکوندا و چند آیتم دیگر که برای کار با مثال‌های ما نیاز دارید صحبت می‌کند.

### مستندات پایتون

هنگام مطالعه این کتاب، مستندات زیر را مفید خواهید یافت:

❖ مرجع زبان پایتون:

❖ <https://docs.python.org/3/reference/index.html>

❖ کتابخانه استاندارد پایتون:

❖ <https://docs.python.org/3/library/index.html>

❖ لیست مستندات پایتون:

❖ <https://docs.python.org/3/>

### پیدا کردن جواب پرسش‌هایتان

انجمن‌های آنلاین معروف پایتون و برنامه‌نویسی عمومی شامل موارد زیر است:

❖ [python-forum.io](http://python-forum.io)

❖ <http://www.dreamincode.net/forums/forum/29-python/>

❖ [StackOverflow.com](https://stackoverflow.com)

همچنین، بسیاری از فروشندگان برای ابزار و کتابخانه‌های خود تالارهای گفتمان ارائه می‌دهند. بسیاری از کتابخانه‌هایی که در این کتاب استفاده خواهید کرد در [github.com](https://github.com) مدیریت و نگهداری می‌شوند. برخی از نگه‌دارندگان کتابخانه از طریق برگه Issues در صفحه

GitHub کتابخانه‌های مزبور را پشتیبانی می‌کنند. اگر نمی‌توانید جوابی برای سؤالات خود به صورت آنلاین پیدا کنید، لطفاً به صفحه وب ما برای کتاب در لینک ذیل مراجعه کنید:

❖ <http://www.deitel.com>

وبسایت ما در حال ارتقا اساسی است. اگر جواب سؤالتان را پیدا نکردید، لطفاً مستقیم به آدرس [deitel@deitel.com](mailto:deitel@deitel.com) ایمیل ارسال کنید.

### دریافت راهنمای JUPYTER

پشتیبانی از نوت‌بوک‌های Jupyter از طریق لینک‌های ذیل قابل دسترسی است:

❖ پروژه Jupyter گروه گوگل:

❖ <https://groups.google.com/forum/#!forum/jupyter>

❖ اتاق گفتگوی Jupyter به صورت بلادرنگ:

❖ <https://gitter.im/jupyter/jupyter>

❖ گیت‌هاب:

❖ <https://github.com/jupyter/help>

❖ StackOverflow:

❖ <https://stackoverflow.com/questions/tagged/jupyter>

❖ Jupyter برای آموزش گروه گوگل (برای آموزش مریدان با Jupyter):

❖ <https://groups.google.com/forum/#!forum/jupyter-education>

### الحاقات

برای استفاده بیشتر از مطالب ارائه شده، باید کد هر مثال را به موازات خواندن بحث مربوطه در کتاب، اجرا کنید. در صفحه وب کتاب در آدرس <http://www.deitel.com> منابع ذیل را برای شما فراهم کرده‌ایم:

❖ کدهای منبع پایتون (فایل‌های .py) و نوت‌بوک‌های Jupyter (فایل‌های .ipynb). قابل دانلود هستند.

❖ فیلم‌های شروع به کار، نحوه استفاده از مثال‌های کد با IPython و نوت‌بوک‌های Jupyter را نشان می‌دهد؛ علاوه بر این در بخش ۱،۵ این ابزار را معرفی می‌کنیم.

❖ پست‌های وبلاگ و به‌روزرسانی‌های کتاب.

برای دستورالعمل‌های دانلود، به بخش **قبل از شروع** که بعد از مقدمه می‌آید، مراجعه کنید.

### ارتباط با نویسندگان

برای پاسخ به سؤالات یا گزارش خطا، یک ایمیل برای ما ارسال کنید:

❖ [deitel@deitel.com](mailto:deitel@deitel.com)

یا از طریق رسانه‌های اجتماعی با ما ارتباط برقرار کنید:

❖ Facebook® (<http://www.deitel.com/deitelfan>)

❖ Twitter® (@deitel)

❖ LinkedIn® (<http://linkedin.com/company/deitel-&-associates>)

❖ YouTube® (<http://youtube.com/DeitelTV>)

### قدردانی‌ها

دوست داریم از باربارا دایتل برای ساعت‌های طولانی که به تحقیقات اینترنتی درباره این پروژه اختصاص داده، تشکر کنیم. خیلی خوشحالیم که با تیم حرفه‌ای متخصصان انتشارات پیرسون همکاری کردیم. از تلاش‌ها و راهنمایی‌های دوست ۲۵ ساله و همکارمان، نایب‌رئیس گروه حرفه‌ای فناوری اطلاعات پیرسون، مارک ال تانوب قدردانی می‌کنیم. مارک و تیم او کتاب‌های حرفه‌ای، محصولات ویدئویی LiveLessons و مسیرهای یادگیری ما را در سرویس Safari منتشر می‌کنند (<https://learning.oreilly.com>). علاوه بر این آن‌ها از سمینارهای آموزش آنلاین Safari ما حمایت مالی می‌کنند. جولی ناهیل تولید این کتاب را مدیریت کرد. ما طرح جلد را انتخاب کردیم و چوتی پرازرتسیث جلد آن را طراحی کرد.

می‌خواهیم از تلاش ارزیاب‌ها و منتقدین خود قدردانی کنیم. پاتریشیا بیرون کیمبال و مگان جاکوبی ارزیاب‌ها را استخدام کردند و روند بررسی را مدیریت کردند. ارزیاب‌ها با رعایت یک برنامه فشرده، کار ما را موشکافی کردند و پیشنهادهای بی‌شماری را برای بهبود دقت، کامل بودن و به‌موقع بودن کار ما ارائه دادند.

### ارزیاب‌ها و منتقدین

دانیل چن، دانشمند علوم داده، تحلیل‌گر لندر	گرت دانسیک، دانشیار علوم کامپیوتر / بیوانفورماتیک، دانشگاه ایالتی کانکتیکات شرقی
پرائشو گوپتا، استادیار، علوم کامپیوتر، دانشگاه دیسیلز	دیوید کوپ، استادیار، مدیر برنامه علوم داده، دانشگاه دارتموث
رامون MataToledo، استاد علوم کامپیوتر، دانشگاه جیمز مدیسون	شیامال میترا، دانشیار، علوم کامپیوتر، دانشگاه تگزاس در آستین
لیسون سانچز، استادیار اقتصاد، دانشگاه سن دیگو	خوزه آنتونیو گونزالس سکو، مشاور فناوری اطلاعات
جیمی ویتاگر، مشاور مستقل علوم داده	الیزابت ویکز، مدرس، دانشکده علوم اطلاعات، دانشگاه ایلینوی
دکتر ایرن برونو، دانشیار گروه علوم و فناوری اطلاعات، دانشگاه جورج میسون	لنس برایانت، دانشیار، گروه ریاضیات، دانشگاه شینسبورگ
دکتر مارشا دیویس، رئیس گروه علوم ریاضی، دانشگاه ایالتی کانکتیکت آمریکا	رولند دپراتی، استاد راهنمای علوم کامپیوتر، دانشگاه ایالتی کانکتیکت آمریکا
شیامال میترا، مدرس ارشد، علوم کامپیوتر، دانشگاه تگزاس در آستین	دکتر مارک پاولی، محقق ارشد، بیوانفورماتیک، دانشکده انفورماتیک میان رشته‌ای، دانشگاه نبراسکا در اوماها
شان رالی، دانشیار ریاضیات، کرسی علوم داده، کالج وست مینستر	آلیسون سانچز، استادیار اقتصاد، دانشگاه سن دیگو
دکتر هاروی سی، دانشیار علوم کامپیوتر، علوم و فناوری اطلاعات، دانشگاه نبراسکا در اوماها	جیمی ویتاگر، مشاور مستقل علوم داده

هنگام مطالعه کتاب، از نظرات، انتقادات، اصلاحات و پیشنهادات شما برای بهبود قدردانی می‌کنیم. لطفاً کلیه مکاتبات را به این آدرس ارسال کنید: [deitel@deitel.com](mailto:deitel@deitel.com). بلافاصله پاسخ خواهیم داد.

دوباره به دنیای جذاب برنامه‌نویسی منبع باز پایتون خوش آمدید. امیدواریم از توسعه برنامه‌های محاسباتی پیشرفته با پایتون، IPython، نوت‌بوک‌های Jupyter، علم داده، هوش مصنوعی، کلان داده و ابر لذت ببرید. برای شما آرزوی موفقیت بزرگ داریم!

پل و هاروی دایتل

### درباره نویسندگان

**پل جی دایتل**، مدیرعامل و مدیر ارشد فنی شرکت دایتل و شرکا، فارغ‌التحصیل MIT است و ۳۸ سال سابقه کار در زمینه کامپیوتر دارد. پل یکی از باتجربه‌ترین مربیان زبان برنامه‌نویسی جهان است که از سال ۱۹۹۲ دوره‌های حرفه‌ای را به توسعه‌دهندگان نرم‌افزار آموزش داده است. او صدها دوره برنامه‌نویسی را به مشتریان صنعت در سطح بین‌المللی ارائه داده است، از جمله سیسکو، IBM، زیمنس، سان مایکروسیستم (اکنون اوراکل)، دل، فیدلیتی، ناسا در مرکز فضایی کندی، آزمایشگاه ملی طوفان شدید، White Sands Missile Range، نرم‌افزار Rogue Wave، بوئینگ، شبکه‌های Nortel، Puma، iRobot و بسیاری دیگر. او و همکارش، دکتر هاروی ام. دایتل، مؤلف پر فروش‌ترین کتاب‌های زبان برنامه‌نویسی درسی / حرفه‌ای / ویدئو هستند.

**دکتر هاروی ام. دایتل**، رئیس و مدیر ارشد استراتژی شرکت دایتل و شرکا، ۵۸ سال تجربه در زمینه کامپیوتر دارد. دکتر دایتل دوره کارشناسی و کارشناسی ارشد خود را در رشته مدرک مهندسی برق از MIT گذرانیده و مدرک دکترای ریاضی خود را از دانشگاه بوستون اخذ کرده است. او قبل از اینکه رشته‌های علوم کامپیوتر مستقل شوند، در هر یک از این مقطع محاسبات را مطالعه می‌کرد. وی قبل از تأسیس شرکت دایتل و شرکا در سال ۱۹۹۱ با پسرش پاول، تجربه گسترده تدریس در کالج، از جمله رئیس گروه علوم کامپیوتر در کالج بوستون را دارد. آثار دایتل به بیش از ۱۰۰ زبان نظیر ژاپنی، آلمانی، روسی، اسپانیایی، فرانسوی، لهستانی، ایتالیایی، چینی ساده‌شده، چینی سنتی، کره‌ای، پرتغالی، یونانی، اردو و ترکی به رسمیت ترجمه شده‌اند. دکتر دایتل صدها دوره برنامه‌نویسی را به مشتریان دانشگاهی، شرکت‌ها، دولتی و نظامی ارائه داده است.

### **درباره شرکت دایتل و شرکا**

شرکت دایتل و شرکا توسط پل جی دایتل و هاروی ام. دایتل تأسیس شد. این شرکت، یک سازمان بین‌المللی در زمینه تألیف و آموزش در حوزه زبان‌های برنامه‌نویسی، فناوری شیء، توسعه برنامه‌های تلفن همراه و فناوری نرم‌افزار اینترنتی و وب است. مشتریان آموزش این شرکت شامل برخی از بزرگ‌ترین شرکت‌های جهان، سازمان‌های دولتی، شاخه‌های نظامی و مؤسسات دانشگاهی است. این شرکت دوره‌های آموزشی برای مریبان در زمینه زبان‌های برنامه‌نویسی مهم در سراسر جهان ارائه می‌دهد.

دایتل و همکارانش، از طریق همکاری ۴۴ ساله انتشار با پیرسون / پرینتیس هال، کتب درسی برنامه‌نویسی و کتاب‌های حرفه‌ای را به صورت کاغذی و الکترونیکی، دوره‌های ویدیویی LiveLessons (برای خرید در <https://www.informit.com> موجود است)، مسیرهای یادگیری و سمینارهای آموزش به صورت آنلاین در سرویس سافاری (<https://learning.oreilly.com>) و دوره‌های چندرسانه‌ای تعاملی Revel™ ارائه می‌دهند.

برای تماس با شرکت دایتل و شرکا، نویسندگان، یا تقاضای پیشنهاد در سایت، آموزش مریبان به نشانی [deitel@deitel.com](mailto:deitel@deitel.com) ایمیل ارسال کنید.

برای کسب اطلاعات بیشتر در مورد آموزش در سایت دایتل به آدرس <http://www.deitel.com/training> مراجعه کنید.

افرادی که مایل به خرید کتاب‌های دایتل هستند می‌توانند خرید خود را در <https://www.amazon.com> انجام دهند.

سفارش‌ها عمده توسط شرکت‌ها، دولت، نهادهای نظامی و دانشگاهی باید مستقیماً با پیرسون انجام گیرد. برای اطلاعات بیشتر، به

آدرس ذیل مراجعه کنید:

❖ <http://www.informit.com/store/sales.aspx>

# قبل از اینکه شروع کنید

این بخش شامل اطلاعاتی است که باید قبل از استفاده از این کتاب مرور کنید. به روزرسانی‌ها را به آدرس ذیل ارسال خواهیم کرد:

❖ <http://www.deitel.com>

## قراردادهای FONT و NAMING

کد، دستورات پایتون، نام فایل‌ها و پوشه‌ها را با فونت sans-serif و اجزای روی صفحه، مانند نام منو را با فونت sans-serif ضخیم نشان می‌دهیم. برای تأکید از فونت ایتالیک و گاهی اوقات برای تأکید بیشتر از فونت ضخیم استفاده می‌کنیم.

## دریافت مثال‌های کد

شما می‌توانید فایل samples.zip حاوی مثال‌های کتاب پایتون برای برنامه‌نویسان را از صفحه وب ما در لینک ذیل دانلود کنید:

❖ <http://www.deitel.com>

روی لینک Download Examples کلیک کنید تا فایل مربوطه در کامپیوتر محلی شما ذخیره شود. بیشتر مرورگرهای وب، فایل را در پوشه Downloads حساب کاربری شما قرار می‌دهند. وقتی دانلود کامل شد، این فایل را در سیستم خود پیدا کنید و پوشه examples را در پوشه Documents حساب کاربری خود باز کنید (از حالت فشرده):

❖ برای ویندوز: \Documents\examples حساب کاربری شما C:\Users\

❖ برای لینوکس یا macOS ~/Documents/examples

اکثر سیستم‌عامل‌ها دارای یک ابزار داخلی برای باز کردن فایل‌های فشرده یا آرشیو هستند. ممکن است بخواهید از یک ابزار آرشیو مانند V-Zip (www.v-zip.org) یا WinZip (www.winzip.com) استفاده کنید.

## ساختار پوشه EXAMPLES

شما سه نوع مثال در این کتاب اجرا خواهید کرد:

❖ کد جداگانه در اسنپت در محیط تعاملی IPython.

❖ برنامه‌های کامل که به‌عنوان اسکریپت شناخته می‌شوند.

❖ نوت‌بوک‌های Jupyter - یک محیط تعاملی مناسب و مبتنی بر وب است که می‌توانید در آن کدی بنویسید، کدی اجرا کنید و کد را با متن، عکس و فیلم ترکیب کنید.

هر یک از این موارد را در قسمت تست‌های بخش ۵-۱ نشان می‌دهیم. در پوشه EXAMPLES به ازای هر فصل یک زیرپوشه وجود دارد. این پوشه‌ها با فرمت ch## نام‌گذاری شده‌اند، ## یک عدد دورقمی از ۰۱ تا ۱۶ بوده و معرف فصل مربوطه می‌باشد - به‌عنوان مثال، ch۰۱. به‌جز فصل‌های ۱۳، ۵ و ۶، پوشه هر فصل شامل موارد زیر است:

❖ snippets\_ipynb - فایل‌های نوت‌بوک Jupyter فصل مربوطه در این پوشه قرار می‌گیرند.

❖ snippets\_py - فایل‌های مربوط به کد منبع پایتون در این پوشه قرار می‌گیرند. اسنپتی که نشان می‌دهیم با یک خط خالی از اسنپت بعدی جدا می‌شود. می‌توانید این اسنپت‌ها را در IPython یا در نوت‌بوک‌های جدید Jupyter که ایجاد می‌کنید کپی و پیست<sup>۰</sup> کنید.

❖ فایل‌های اسکریپت و فایل‌های پشتیبانی‌کننده آن‌ها.

فصل ۱۳ شامل یک برنامه کاربردی است. در فصل ۱۵ و ۱۶ توضیح داده می‌شود که چگونه فایل‌های موردنیاز خود را در پوشه‌های ch۱۵ و ch۱۶ پیدا کنید.

## نصب آناکوندا<sup>۱</sup>

در این کتاب از توزیع آسان آناکوندا پایتون استفاده می‌کنیم. این نسخه با تمام مواردی که برای کار با مثال‌های ما نیاز دارید، ارائه می‌شود، برای مثال:

❖ مفسر IPython،

<sup>۰</sup> paste  
<sup>۱</sup> ANACONDA

❖ بیشتر کتابخانه‌ها پایتون و کتابخانه‌های علوم داده که استفاده می‌کنیم،  
❖ یک سرور محلی نوت‌بوک‌های Jupyter و بنابراین می‌توانید نوت‌بوک‌های ما را بارگیری و اجرا کنید و  
❖ بسته‌های نرم‌افزاری مختلف دیگری مانند (IDE) اسپایدر<sup>۷</sup> - که البته در این کتاب فقط از نوت‌بوک‌های IPython و Jupyter استفاده می‌کنیم.

برنامه نصب کننده ۸ آناکوندا برای Python ۳.x را برای ویندوز، macOS یا لینوکس از لینک ذیل دانلود کنید:

❖ <https://www.anaconda.com/download/>

پس از اتمام دانلود، نصب کننده را اجرا کرده و دستورالعمل‌های روی صفحه‌نمایش را دنبال کنید. برای اطمینان از اینکه آناکوندا به درستی اجرا می‌شود، پس از نصب، فایل‌ها آن را جابجا نکنید.

## به‌روزرسانی آناکوندا

در ادامه، اطمینان حاصل کنید که آناکوندا به‌روز است. برای این کار، یک پنجره خط فرمان در سیستم خود باز کنید:

❖ در macOS، یک ترمینال را از زیرپوشه Utilities در پوشه Applications باز کنید.

❖ در ویندوز، Anaconda Prompt را از منوی start باز کنید. هنگام انجام این کار برای به‌روزرسانی Anaconda (همان‌طور که در اینجا انجام خواهید داد) یا نصب بسته‌های جدید (چند لحظه پیش بررسی شد)، Anaconda Prompt را به‌عنوان administrator اجرا کنید. برای این کار روی آن راست کلیک کرده، سپس گزینه More > Run as administrator را انتخاب کنید. (اگر Anaconda را پیدا نکردید در قسمت type here to search در پایین صفحه منوی start نام آن را تایپ کنید و به‌راحتی آن را پیدا کنید.)

❖ در لینوکس، ترمینال یا shell سیستم خود را باز کنید (بسته به توزیع لینوکس متفاوت است).

در پنجره خط فرمان سیستم خود، دستورات زیر را برای به‌روزرسانی بسته‌های نصب شده آناکوندا به آخرین نسخه‌ها اجرا کنید:

```
۱. conda update conda
۲. conda update --all
```

## PACKAGE MANAGER

دستور conda استفاده شده در بالا package manager را فراخوانی می‌کند و در حقیقت یکی از دو package manager اصلی پایتون است که در این کتاب استفاده خواهید کرد. package manager دیگر pip است. بسته‌ها حاوی فایل‌های موردنیاز برای نصب یک کتابخانه یا ابزار خاص پایتون هستند. در کل کتاب، برای نصب بسته‌های اضافی از conda استفاده خواهید کرد، مگر اینکه این بسته‌ها از طریق conda در دسترس نباشند، در این صورت از pip استفاده خواهید کرد. برخی ترجیح می‌دهند از pip استفاده کنند، زیرا در حال حاضر از بسته‌های بیشتری پشتیبانی می‌کند. اگر در نصب بسته‌ای با conda با مشکلی مواجه شدید، pip را امتحان کنید.

## نصب ابزار تحلیل گر استاتیکی کد موسوم به Prospector

ممکن است بخواهید کد پایتون خود را با استفاده از ابزار تجزیه و تحلیل Prospector که کد شما را برای خطاهای رایج بررسی می‌کند و در بهبود آن به شما کمک می‌کند، آنالیز کنید. برای نصب Prospector و کتابخانه‌های پایتونی که استفاده می‌کند، دستور زیر را در پنجره خط فرمان اجرا کنید:

❖ `pip install prospector`

## نصب JUPYTER-MATPLOTLIB

چندین انیمیشن را با استفاده از کتابخانه رسم نمودار و گرافیک به نام matplotlib پیاده‌سازی می‌کنیم. برای استفاده از این انیمیشن‌ها در نوت‌بوک‌های Jupyter، باید ابزاری به نام ipympl نصب کنید. در ترمینال، خط فرمان یا shell آناکوندا که قبلاً باز کرده‌اید، دستورات زیر را اجرا کنید (در هر لحظه یک فرمان):

```
conda install -c conda-forge ipympl conda install
nodejs
jupyter labextension install @jupyter-widgets/jupyterlab-manager
jupyter labextension install jupyter-matplotlib
https://github.com/matplotlib/jupyter-matplotlib
```

<sup>۷</sup> Spyder  
<sup>۸</sup> installer  
<sup>۹</sup> package

## نصب سایر بسته‌ها

آناکوندا تقریباً ۳۰۰ بسته معروف علمی پایتون و علوم داده را برای شما فراهم می‌کند، برای مثال NumPy، matplotlib، pandas، BeautifulSoup، Requests، Bokeh، SciPy، SciKitLearn، Seaborn، spacy، sqlalchemy، مدل‌های stats و بسیاری موارد دیگر مثال‌هایی از این دست محسوب می‌شوند. تعداد بسته‌های اضافی که باید در سراسر کتاب نصب کنید کم خواهند بود و در صورت لزوم دستورالعمل نصب را ارائه می‌دهیم. با بررسی بسته‌های جدید، مستندات آن‌ها نحوه نصب آن‌ها را توضیح می‌دهند.

## اخذ حساب کاربری توسعه‌دهنده در توییتر

اگر قصد دارید از فصل داده کاوی توییتر و سایر مثال‌های مبتنی بر توییتر در فصل‌های بعدی استفاده کنید، برای یک حساب توسعه‌دهنده توییتر درخواست کنید. در زمان نگارش این کتاب برای دسترسی به API های توییتر نیاز به ثبت‌نام دارید. برای درخواست، فرم درخواست را در آدرس <https://developer.twitter.com/en/apply-for-access> پر کنید و ارسال کنید. توییتر هر درخواستی را بررسی می‌کند. در زمان نگارش این کتاب، حساب‌های توسعه‌دهنده شخصی بلافاصله ایجاد می‌شدند و درخواست‌های حساب کاربری شرکت‌ها از چند روز تا چند هفته طول می‌کشیدند. در ضمن هیچ ضمانتی برای تصویب درخواست وجود ندارد.

## برخی از فصل‌ها اتصال به اینترنت نیاز است

در حین استفاده از این کتاب، برای نصب چندین کتابخانه اضافی پایتون به اتصال اینترنت احتیاج دارید. در بعضی از فصل‌ها، برای حساب‌هایی که برای سرویس‌های مبتنی بر ابر ثبت‌نام می‌کنید و عمدتاً برای استفاده از اکانت‌های رایگان آن‌ها هم هستند به اتصال اینترنت نیاز دارید. برخی از سرویس‌ها برای تأیید هویت شما به کارت‌های اعتباری نیاز دارند. در چند مورد، از سرویس‌هایی که رایگان نیستند استفاده خواهید کرد. در این موارد، از اعتبارات پولی ارائه‌شده توسط فروشندگان استفاده خواهید کرد، بنابراین می‌توانید خدمات آن‌ها را بدون پرداخت هزینه امتحان کنید. احتیاط: برخی از سرویس‌های مبتنی بر ابر که راه‌اندازی و تنظیم می‌کنید هزینه‌هایی را به شما تحمیل خواهند کرد. وقتی مطالعات موردی ما را با استفاده از چنین سرویس‌های به پایان رسانید، حتماً منابعی را که اختصاص داده‌اید فوراً حذف کنید.

## تفاوت‌های کوچک در خروجی‌های برنامه

وقتی مثال‌هایتان را اجرا می‌کنید، ممکن است تفاوت‌هایی کوچکی بین نتایج نشان داده‌شده و نتایج خود مشاهده کنید:

- ❖ به دلیل تفاوت در نحوه انجام محاسبات اعداد اعشاری (مانند -۱۲۳,۴۵، ۷,۵ یا ۰,۲۳۶۹۳۷) در سیستم‌عامل‌ها، ممکن است تغییرات جزئی در خروجی‌ها- به‌ویژه در رقم‌های سمت راست نقطه اعشار مشاهده کنید.
- ❖ وقتی خروجی‌هایی را نشان می‌دهیم که در پنجره‌های جداگانه ظاهر می‌شوند، پنجره‌ها را برش می‌دهیم تا حاشیه‌های آن‌ها برداشته شود.

## پیدا کردن جواب پرسش‌هایتان

تالارهای گفتگوی آنلاین این امکان را به شما می‌دهند تا با سایر برنامه‌نویسان پایتون تعامل داشته باشید و پاسخ سؤال‌هایتان را بگیرید. انجمن‌های آنلاین معروف پایتون و برنامه‌نویسی عمومی شامل موارد زیر هستند:

- ❖ [python-forum.io](http://python-forum.io)
- ❖ <http://www.dreamincode.net/forums/forum/29-python/>
- ❖ [StackOverflow.com](http://StackOverflow.com)

همچنین، بسیاری از فروشندگان برای ابزار و کتابخانه‌های خود تالارهای گفتگو ارائه می‌دهند. بسیاری از کتابخانه‌هایی که در این کتاب استفاده خواهید کرد در [github.com](http://github.com) مدیریت و نگهداری می‌شوند. برخی از نگه‌دارندگان کتابخانه از طریق برگه Issues در صفحه GitHub کتابخانه‌های مزبور را پشتیبانی می‌کنند. اگر نمی‌توانید جوابی برای سؤالات خود به‌صورت آنلاین پیدا کنید، لطفاً به صفحه وب ما برای کتاب در لینک ذیل مراجعه کنید:

- ❖ <http://www.deitel.com>

وب‌سایت ما در حال ارتقا اساسی است. اگر جواب سؤال‌تان را پیدا نکردید، لطفاً مستقیم به آدرس [deitel@deitel.com](mailto:deitel@deitel.com) ایمیل ارسال کنید.

اکنون آماده‌اید تا خواندن پایتون برای برنامه‌نویسان را شروع کنید. برای شما موفقیتی بزرگ آرزو مندیم!



# مقدمه‌ای بر کامپیوترها و پایتون<sup>۱</sup>

## اهداف

### در این فصل:

- ❖ مطالبی در مورد توسعه‌های جدید و هیجان‌انگیز محاسبات کامپیوتری یاد خواهید گرفت.
- ❖ اصول برنامه‌نویسی شیء‌گرا<sup>۲</sup> را مرور می‌کنید.
- ❖ نقاط قوت پایتون را درک خواهید کرد.
- ❖ با کلمات کلیدی پایتون و کتابخانه‌های علوم داده که در این کتاب مورد استفاده قرار می‌گیرد، آشنا می‌شوید.
- ❖ برای اجرای کدهای پایتون<sup>۳</sup> به‌طور آزمایشی مفسر IPython را در وضعیت تعاملی اجرا می‌کنید.
- ❖ برای رسم نمودار میله‌ای متحرک یک اسکریپت<sup>۴</sup> پایتون اجرا می‌کنید.
- ❖ ایجاد و اجرای تستی برنامه تحت وب Jupyter Notebook برای اجرای کدهای پایتون.
- ❖ یاد می‌گیرید که کلان داده‌ها<sup>۵</sup> تا چه بزرگ هستند و با چه سرعتی رشد می‌کنند.
- ❖ به‌عنوان یک مطالعه موردی درباره کلان داده یک برنامه مشهور مسیریاب تلفن همراه بررسی قرار می‌گیرد.
- ❖ هوش مصنوعی به شما معرفی می‌گردد، این موضوع فصل اشتراک علوم کامپیوتر و علوم داده می‌شود.

## رئوس مطالب

۱. مقدمه
۲. مرور سریع اصول فناوری شیء
۳. پایتون
۴. کتابخانه‌ها
  - ۴-۱. کتابخانه‌های استاندارد پایتون
  - ۴-۲. کتابخانه‌های علوم داده
۵. اجرای آزمایشی با استفاده از IPython و Jupyter Notebooks
  - ۵-۱. استفاده از IPython در وضعیت محاوره‌ای به‌عنوان یک ماشین حساب
  - ۵-۲. اجرای برنامه پایتون با استفاده از مفسر IPython
  - ۵-۳. نوشتن و اجرای کد با Jupyter Notebook

---

<sup>۱</sup> Python  
<sup>۲</sup> object-oriented programming  
<sup>۳</sup> interactive  
<sup>۴</sup> script  
<sup>۵</sup> big data

۶. رایانش ابری<sup>۶</sup> و اینترنت اشیا<sup>۷</sup>

۶-۱. رایانش ابری

۶-۲. اینترنت اشیا

۷. کلان داده‌ها تا چه حد بزرگ هستند.

۷-۱. تجزیه و تحلیل کلان داده

۷-۲. آیا علوم داده و کلان داده یکی هستند؟ بررسی موارد کاربرد هریک

۸. مطالعه موردی کلان داده در برنامه تلفن همراه

۹. معرفی علم داده. هوش مصنوعی فصل اشتراک علوم کامپیوتر و علم داده

۱۰. جمع‌بندی

## ۱-۱. مقدمه

به برنامه‌نویسی پایتون که یکی از پرکاربردترین زبان‌های برنامه‌نویسی جهان محسوب می‌شود، خوش آمدید. بر اساس شاخص محبوبیت زبان‌های برنامه‌نویسی<sup>۸</sup>، پایتون محبوب‌ترین زبان به شمار می‌رود.

<https://pypl.github.io/PYPL.html>

در این فصل، اصطلاحات و مفاهیمی را معرفی می‌کنیم که پایه و اساس یادگیری برنامه‌نویسی پایتون محسوب می‌شوند. در فصل‌های ۲ تا ۱۰ برنامه‌نویسی و در فصل‌های ۶ تا ۱۱ موارد مطالعاتی بر مبنای داده، هوش مصنوعی و رایانش ابری ارائه می‌شود.

اصطلاحات و مفاهیم برنامه‌نویسی شیء‌گرا را بررسی خواهیم کرد. خواهید آموخت که چرا پایتون تا این حد محبوب شده است. کتابخانه استاندارد پایتون و کتابخانه‌های مختلف علوم داده را معرفی خواهیم کرد و این کتابخانه‌ها به شما کمک می‌کنند از نوشتن مجدد کد جلوگیری کنید. از این کتابخانه‌ها برای ایجاد اشیا نرم‌افزاری استفاده خواهید کرد و با کمک این اشیا کارهای زیادی را با حداقل دستورات انجام خواهید داد.

در ادامه با سه روش اجرای کدهای پایتون آشنا خواهید شد:

❖ در روش اول، از IPython برای اجرای تعاملی دستورالعمل‌های پایتون استفاده کرده و بلافاصله نتایج آن‌ها را مشاهده خواهید کرد.

❖ در روش دوم یک برنامه نسبتاً بزرگ خواهید نوشت و این برنامه خلاصه نتایج ۶ عدد در پرتاب تاس را به صورت نمودار میله‌ای متحرک نشان می‌دهد. در این برنامه با قانون اعداد بزرگ در عمل آشنا خواهید شد. در فصل ۶ این برنامه را با استفاده از کتابخانه Matplotlib visualization ایجاد خواهید کرد.

<sup>۶</sup> Cloud

<sup>۷</sup> Internet of Things

<sup>۸</sup> Popularity of Programming Languages (PYPL)

## مقدمه‌ای بر کامپیوترها و پایتون ۲۷

❖ در آخرین روش Jupyter Notebooks را معرفی خواهیم کرد. با استفاده از upyterLab که یک ابزار تحت وب محسوب می‌شود به راحتی می‌توانید برنامه خود را نوشته و دستورات پایتون را اجرا کنید. Jupyter Notebooks به شما امکان می‌دهد در برنامه خود متن، عکس، صدا، فیلم، انیمیشن و کد بگنجانید.

در گذشته، بیشتر برنامه‌های کامپیوتری روی کامپیوترهای مستقل اجرا می‌شدند (یعنی باهم شبکه نبودند). برنامه‌های امروزی را می‌توان باهدف برقراری ارتباط بین میلیاردها کامپیوتر از طریق اینترنت نوشت. مفاهیم رایانش ابری و اینترنت اشیا را معرفی خواهیم کرد و زمینه لازم برای نوشتن برنامه‌های امروزی در فصول ۶ تا ۱۱ را برای شما ایجاد خواهیم کرد.

در ادامه یاد می‌گیرید که کلان داده‌ها چقدر بزرگ هستند و با چه سرعتی رشد می‌کنند. در مرحله بعدی، یک مطالعه موردی کلان داده‌ها را در مورد برنامه مسیریاب ویز<sup>۹</sup> تلفن همراه ارائه خواهیم داد. این برنامه با استفاده از بسیاری از فن‌آوری‌های مدرن، یک مسیر رانندگی پویا را ارائه می‌دهد که تا با سریع‌ترین زمان و بهترین حالت به مقصد برسد. همان‌طور که در مورد این فن‌آوری‌ها صحبت می‌کنیم، موارد استفاده آن‌ها در این کتاب را ذکر خواهیم کرد. این فصل با اولین آشنایی ما به بخش علوم داده که در حقیقت یک فصل اشتراک کلیدی بین علوم کامپیوتر و هوش مصنوعی محسوب می‌شود، خاتمه می‌یابد.

## ۲-۱. مرور سریع اصول فناوری شیء

از آنجائی که تقاضا برای نرم‌افزارهای جدید و قدرتمند در حال افزایش است، ساختن نرم‌افزار با سرعت، با عملکرد صحیح و البته اقتصادی اهمیت زیادی دارد. اشیا و یا به عبارت دقیق‌تر، کلاس‌هایی که اشیا از آن‌ها ایجاد می‌شوند، اساساً اجزای نرم‌افزاری قابل‌استفاده مجدد هستند. اشیا تاریخ، اشیا زمان، اشیا صدا، اشیا ویدئو، اشیا اتومبیل، اشیا افراد و غیره نمونه‌هایی از اشیا هستند. تقریباً هر اسمی از نظر ویژگی‌ها (به‌عنوان مثال: نام، رنگ و اندازه) و رفتارها (به‌عنوان مثال: محاسبه، جابجایی و برقراری ارتباط) می‌تواند به‌طور معقول به‌عنوان یک شیء نرم‌افزاری نشان داده شود. گروه‌های توسعه‌دهنده نرم‌افزار می‌توانند از رویکرد ماژولار طراحی و پیاده‌سازی شیء گرا استفاده کنند. رویکرد اخیر از روش‌های معروف قبلی مانند برنامه‌نویسی ساخت‌یافته بسیار کارآمدتر می‌باشد. درک، برطرف کردن خطا و اصلاح برنامه‌های شیء گرا اغلب آسان‌تر است.

### اتومبیل به‌عنوان یک شیء

برای کمک به شما در درک اشیا و محتوای آن‌ها، بیایید با یک قیاس ساده شروع کنیم. فرض کنید با اتومبیل رانندگی کرده و با فشار دادن پدال گاز سرعت آن را بیشتر کنید. قبل از انجام این کار چه اتفاقی می‌افتد؟ خوب،

<sup>۹</sup> waze

قبل از راندن یک ماشین، باید یک شخص اتومبیل را طراحی کند. یک اتومبیل به طور معمول با عنوان نقشه‌های مهندسی شبیه نقشه‌های طراحی خانه شروع می‌شود. پدال گاز جزئیات پیچیده مکانیسم افزایش سرعت اتومبیل را از راننده پنهان می‌کند، به همین نحو پدال ترمز و فرمان اتومبیل جزئیات مکانیسم‌های کاهش سرعت و پیچیدن اتومبیل را از راننده پنهان می‌کند. این رویکرد برای افرادی که دانشی کم و یا کسانی که هیچ‌گونه دانشی در مورد موتور، ترمز و فرمان اتومبیل ندارند، این امکان را فراهم می‌کند تا با اتومبیل به راحتی رانندگی کنند.

همان‌طور که نمی‌توانید وعده‌های غذایی را در طرح اولیه آشپزخانه بپذیرید، نمی‌توانید با نقشه‌های مهندسی اتومبیل رانندگی کنید. قبل از اینکه بتوانید با ماشین رانندگی کنید، باید از نقشه‌های مهندسی که ماشین را توصیف می‌کنند یک اتومبیل ساخته شود. یک اتومبیل کامل یک پدال گاز واقعی برای افزایش سرعت دارد، اما حتی این هم کافی نیست و اتومبیل به خودی‌خود شتاب نمی‌گیرد (امیدوارم!) بنابراین راننده باید پدال را فشار دهد تا ماشین شتاب بگیرد.

### متدها و کلاس‌ها

بیایید از مثال اتومبیل برای معرفی برخی مفاهیم کلیدی برنامه‌نویسی شیء گرا استفاده کنیم. انجام یک کار در یک برنامه به یک متد نیاز دارد. در این متد کدهای قرار دارند تا وظیفه آن متد را انجام دهند. همان‌طور که پدال گاز یک ماشین مکانیسم‌های افزایش سرعت اتومبیل را از دید راننده پنهان می‌کند، یک متد این دستورات را از کاربر پنهان می‌کند. در پایتون، یک واحد برنامه به نام کلاس، مجموعه‌ای از متدها دارد که وظایف کلاس را انجام می‌دهند. به عنوان مثال، کلاسی که نشان‌دهنده یک حساب بانکی است ممکن است حاوی یک متد برای واریز پول به حساب، متد دیگری برای برداشت پول از حساب و متد سومی برای تعیین موجودی حساب باشد. یک کلاس از نظر مفهومی مشابه نقشه‌های مهندسی اتومبیل است که طراحی پدال گاز، فرمان و ... را در خود جای داده است.

### نمونه‌سازی<sup>۱۰</sup>

همان‌طور که قبل از راندن یک اتومبیل باید شخصی، از نقشه‌های مهندسی اتومبیل یک اتومبیل بسازد، قبل از اینکه برنامه‌ای بتواند وظایفی را که متدهای کلاس تعریف می‌کنند اجرا کند، باید یک شیء از کلاس ایجاد شود. فرایند انجام این کار را نمونه‌سازی می‌نامند. از آن به بعد به یک شیء به عنوان نمونه‌ای از کلاس مربوطه محسوب می‌شود.

<sup>۱۰</sup> Instantiation

## استفاده مجدد<sup>۱۱</sup>

درست همان‌طور که می‌توان از نقشه‌های مهندسی اتومبیل برای تولید تعداد زیادی از ماشین‌ها استفاده مجدد کرد، شما هم می‌توانید بارها از یک کلاس به‌راحتی برای ساخت بسیاری از اشیا استفاده کنید. استفاده مجدد از کلاس‌های موجود در زمان ساخت کلاس‌ها و برنامه‌های جدید، موجب صرفه‌جویی در زمان و کار می‌شود. از آنجائی که کلاس‌ها و اجزای نرم‌افزاری موجود اغلب تحت **آزمون‌های**<sup>۱۲</sup> زیاد، **عیب‌یابی**<sup>۱۳</sup> و **تنظیم کارایی**<sup>۱۴</sup> قرار گرفته‌اند به شما کمک می‌کنند تا سیستم‌های قابل‌اطمینان و کارآمدتر ایجاد کنید. درست همان‌طور که مفهوم قطعات قابل تعویض برای انقلاب صنعتی بسیار حیاتی بودند، مفهوم کلاس‌های با قابلیت استفاده مجدد برای انقلاب نرم‌افزاری بسیار مهم تلقی می‌شوند و در حقیقت ماحصل فناوری شیء‌گرا هستند.

## پیام‌ها و فراخوانی متدها

هنگامی که حین رانندگی با اتومبیل پدال گاز را فشار می‌دهید، در حقیقت پیامی را برای انجام یک وظیفه و یا افزایش سرعت به اتومبیل ارسال می‌کنید. به همین ترتیب، به یک شیء پیام ارسال می‌کنید. هر پیام به‌صورت فراخوانی یک متد اجرا می‌شود و به متد آن شیء می‌گوید تا وظیفه خود را انجام دهد. به‌عنوان مثال، ممکن است یک برنامه برای افزایش موجودی شیء bank-account، متد deposit آن را فراخوانی کند.

## صفت‌ها و متغیرهای نمونه‌سازی شده

یک اتومبیل علاوه بر داشتن قابلیت‌هایی برای انجام وظایف خود، دارای ویژگی‌هایی نظیر رنگ، تعداد در، مقدار سوخت موجود در باک، سرعت فعلی آن و ثبت کل مسافت طی شده (به‌عنوان مثال، خواندن کیلومتر شمار) است. ویژگی‌های خودرو مانند قابلیت‌های آن به‌عنوان بخشی از طراحی خودرو در نقشه‌های مهندسی نشان داده می‌شوند (به‌عنوان مثال شامل کیلومتر شمار و شاخص سوخت است). علاوه بر این، هنگامی که با اتومبیل واقعی رانندگی می‌کنید، این ویژگی‌ها همراه با ماشین حمل می‌شوند. هر اتومبیل ویژگی‌های خاص خود را حفظ می‌کند. به‌عنوان مثال، هر اتومبیل می‌داند که در باک خود چه مقدار سوخت دارد، اما در مورد مقدار سوخت موجود در باک سایر اتومبیل‌ها چیزی نمی‌داند.

به همین نحو یک شیء دارای ویژگی‌هایی است که همراه با آن در یک برنامه استفاده می‌شود. این ویژگی‌ها به‌عنوان بخشی از کلاس شیء مشخص می‌شوند. برای مثال، یک شیء حساب بانکی، دارای ویژگی موجودی است که مقدار پول در آن حساب را نشان می‌دهد. هر شیء حساب بانکی موجود خود را می‌داند، اما دو مورد موجودی

---

<sup>۱۱</sup> Reuse

<sup>۱۲</sup> testing

<sup>۱۳</sup> debugging

<sup>۱۴</sup> performance tuning

دیگر حساب‌ها چیزی نمی‌داند. این ویژگی‌ها توسط متغیرهای نمونه‌سازی شده کلاس مشخص می‌شوند. ویژگی‌ها و متدهای یک کلاس (و اشیای آن) ارتباط تنگاتنگی با یکدیگر دارند و به همین دلیل کلاس‌ها ویژگی‌ها و متدهای خود را بسته‌بندی می‌کنند.

## وراثت<sup>۱۵</sup>

با استفاده از وراثت به راحتی می‌توان یک کلاس جدید (به نام کلاس **فرزند**<sup>۱۶</sup>) را ایجاد کرد. این فرآیند با ویژگی‌های یک کلاس موجود (به نام کلاس **والد**<sup>۱۷</sup>) شروع شده، احتمالاً آن‌ها را شخصی‌سازی کرده و ویژگی‌های منحصربه‌فرد خود را با آن اضافه می‌کند. در مثال اتومبیل، مطمئناً یک شیء از کلاس convertible یک شیء از کلاس عمومی اتومبیل است، اما به‌طور خاص‌تر، سقف این خودرو قابلیت باز و بسته شدن دارد.

## تحلیل و طراحی شیء‌گرا<sup>۱۸</sup>

به‌زودی برنامه‌هایی به زبان پایتون خواهید داشت اما چگونه کد برنامه‌های خود را آماده می‌کنید؟ شاید مانند بسیاری از برنامه نویسان به‌سادگی کامپیوتر خود را روشن کرده و شروع به تایپ کردن کنید. ممکن است این روش برای برنامه‌های کوچک (مانند برنامه‌هایی که در فصل‌های اول کتاب ارائه شده‌اند) کارساز باشد، اما اگر از شما خواسته شد یک سیستم نرم‌افزاری ایجاد کنید تا هزاران دستگاه خودپرداز را برای یک بانک بزرگ کنترل کند؟ یا فرض کنید که از شما خواسته شد روی گروهی متشکل از ۱۰۰۰ توسعه‌دهنده نرم‌افزار کار کنید که تا نسل بعدی سیستم کنترل ترافیک هوایی ایالت متحده را بسازند؟ برای پروژه‌های بزرگ و پیچیده، نباید به‌سادگی بنشینید و شروع به نوشتن برنامه کنید.

برای ایجاد بهترین راه‌حل‌ها، باید یک فرایند تجزیه و تحلیل دقیق برای تعیین نیازمندی‌های پروژه خود دنبال کنید (به‌عنوان مثال، تعیین کنید که قرار است سیستم چه کاری انجام دهد)، سپس طراحی را تهیه کنید که این نیازمندی‌ها را برآورده می‌کند (به‌عنوان مثال، مشخص کنید که سیستم چگونه باید این کار را انجام دهد). در حالت ایدئال، می‌بایست قبل از نوشتن هر کد، این مراحل را طی کرده و با دقت طراحی را بررسی کنید (و طرح خود را با متخصصان دیگر نرم‌افزار هم مورد بررسی قرار دهید). اگر این فرایند شامل تجزیه و تحلیل و طراحی سیستم از دیدگاه شیء‌گرا باشد، فرآیند تحلیل و طراحی شیء‌گرا (OOAD) نامیده می‌شود. زبان‌هایی مانند پایتون شیء‌گرا هستند. برنامه‌نویسی در چنین زبانی که برنامه‌نویسی شیء‌گرا (OOP) نامیده می‌شود، به شما امکان می‌دهد یک طرح شیء‌گرا را به‌عنوان یک سیستم کاری پیاده‌سازی کنید.

<sup>۱۵</sup> Inheritance

<sup>۱۶</sup> subclass

<sup>۱۷</sup> superclass

<sup>۱۸</sup> Object-Oriented Analysis and Design (OOAD)

## ۳-۱. پایتون

پایتون یک زبان برنامه‌نویسی است که در سال ۱۹۹۱ صورت عمومی منتشر شد. این زبان توسط **خودو فان روسوم**<sup>۱۹</sup> از موسسه ملی تحقیقات ریاضیات و علوم کامپیوتر آمستردام توسعه یافت. زبان پایتون به یکی از رایج‌ترین زبان‌های برنامه‌نویسی دنیا تبدیل شده است. این زبان در حال حاضر به‌طور خاصی برای مصارف آموزشی و علمی محبوب است و به‌تازگی از زبان برنامه‌نویسی R به‌عنوان محبوب‌ترین زبان برنامه‌نویسی پیشی گرفته است. در اینجا دلایلی چند وجود دارد که چرا پایتون محبوب است و همه باید آن را یاد بگیرند.

<https://www.oreilly.com/ideas/5-things-to-watch-in-python-in-2017>.  
<https://www.kdnuggets.com/2017/08/python-overtakes-r-leader-nalytics-data-science.html>.  
<https://www.r-bloggers.com/data-science-job-report-2017-r-passes-as-but-python-leaves-them-both-behind>  
<https://www.oreilly.com/ideas/5-things-to-watch-in-python-in-2017>.  
<https://dbader.org/blog/why-learn-python>.  
<https://simpleprogrammer.com/2017/01/18/7-reasons-why-you-should-earn-python/>.  
<https://www.oreilly.com/ideas/5-things-to-watch-in-python-in-2017>.

- ❖ این زبان **متن‌باز**<sup>۲۰</sup> و رایگان بوده و به‌طور گسترده‌ای از طریق انجمن‌های متن‌باز قابل دسترسی است.
- ❖ یادگیری این زبان نسبت به زبان‌های C، C++، C# و جاوا راحت‌تر بوده و این امکان را به توسعه‌دهنده‌ای مبتدی و حرفه‌ای می‌دهد تا سریع‌تر راه بیفتند.
- ❖ این زبان در قیاس با سایر زبان‌ها از خوانایی بیشتری برخوردار است.
- ❖ به‌طور گسترده‌ای در آموزش استفاده می‌شود.
- ❖ این زبان با کتابخانه‌های استاندارد گسترده و کتابخانه‌های منبع باز ثالث، بهره‌وری توسعه‌دهنده را افزایش می‌دهد، به همین دلیل برنامه‌نویسان می‌توانند سریع‌تر کد بنویسند و کارهای پیچیده را با حداقل کد انجام دهند. در این مورد در بخش ۱-۴ بیشتر صحبت خواهیم کرد.
- ❖ برنامه‌های متن‌باز و رایگان زیادی به زبان پایتون وجود دارد.
- ❖ در توسعه وب محبوب است (برای مثال Django و Flask).
- ❖ این زبان از الگوی برنامه‌نویسی رایج نظیر رویه‌ای، سبک تابعی و شیء‌گرا پشتیبانی می‌کند. در فصل ۴ معرفی ویژگی‌های برنامه‌نویسی تابعی را شروع می‌کنیم و در فصل‌های بعدی از آن‌ها استفاده می‌کنیم.

[https://en.wikipedia.org/wiki/Python\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))

<sup>۱۹</sup> Guido van Rossum

<sup>۲۰</sup> Open source

❖ این زبان برنامه‌نویسی **هم‌زمانی**<sup>۲۱</sup> را با استفاده از `async/await` و `asyncio` آسان کرده و به راحتی می‌توانید کدهای هم‌زمانی **تک نخه**<sup>۲۲</sup> بنویسید. این زبان فرایندهای پیچیده ذاتی کد نویسی، اشکال‌زدایی و نگهداری چنین برنامه‌های را آسان می‌کند.

<https://docs.python.org/3/library/asyncio.html>.

<https://www.oreilly.com/ideas/5-things-to-watch-in-python-in-017>.

❖ قابلیت‌های زیادی برای افزایش کارایی پایتون وجود دارد.

❖ از این زبان برای نوشتن اسکریپت‌های ساده تا برنامه‌های پیچیده‌ای با تعداد زیادی کاربر نظیر YouTube، Reddit، Dropbox و Quora استفاده می‌شود.

[https://www.hartmannsoftware.com/Blog/Articles\\_from\\_Software\\_Fans/Most\\_Amoumous-Software-Programs-Written-in-Python](https://www.hartmannsoftware.com/Blog/Articles_from_Software_Fans/Most_Amoumous-Software-Programs-Written-in-Python).

❖ این زبان در شاخه هوش مصنوعی که در حال رشد انفجاری است محبوبیت زیادی دارد. بخشی از آن به دلیل ارتباط ویژه آن با علم داده است.

❖ پایتون به طور گسترده‌ای در مجامع مالی استفاده می‌شود.

❖ بازار کار گسترده‌ای برای برنامه‌نویسان پایتون در بسیاری از شاخه‌ها به‌ویژه در موقعیت‌های مرتبط با علوم داده وجود دارد و برنامه‌نویسان پایتون بالاترین دریافتی را دارند.

<https://www.infoworld.com/article/3170838/developer/get-paid-10-programming-languages-to-learn-in-2017.html>

<https://medium.com/@ChallengeRocket/top-10-of-programming-languages-with-the-highest-salaries-in-2017-4390f468256e>.

❖ R یک زبان برنامه‌نویس متن‌باز محبوب برای برنامه‌های آماری و تصویرسازی است. پایتون و R دو زبان بسیار گسترده علوم داده هستند.

## توزیع آناکوندا<sup>۲۳</sup> پایتون

از توزیع آناکوندا پایتون استفاده می‌کنیم زیرا نصب آن در ویندوز، macOS و Linux آسان است و از آخرین نسخه‌های Python، یعنی مفسر IPython (در بخش ۱-۵-۱) معرفی می‌شود) و Jupyter Notebook (در بخش ۱-۵-۳) معرفی می‌شود) پشتیبانی می‌کند. علاوه بر این، آناکوندا شامل بسته‌های نرم‌افزاری و کتابخانه‌های دیگری است که معمولاً در برنامه‌نویسی پایتون و علم داده، به شما اجازه می‌دهد تا به جای مسائل نصب نرم‌افزار، روی پایتون و علم داده تمرکز کنید. مفسر IPython دارای ویژگی‌هایی است که به شما کمک می‌کند پایتون، کتابخانه استاندارد پایتون و مجموعه گسترده کتابخانه‌های شخص ثالث را کشف، تجربه و آزمایش کنید.

<sup>۲۱</sup> concurrent

<sup>۲۲</sup> single-threaded

<sup>۲۳</sup> Anaconda



<https://ipython.org>

## ذن پایتون<sup>۲۴</sup>

ما به ذن پایتون پایبند هستیم، ذن پایتون اصولی است که توسط خالق این زبان ایجاد شده و توسط تیم پیترز<sup>۲۵</sup> نوشته شده است. با اجرای دستور `import this` در محیط IPython می‌توان این لیست را مشاهده کرد. ذن پایتون در پروپوزال **بهبودسازی پایتون**<sup>۲۶</sup> و یا PEP تعریف شده است. PEP یک سند طراحی است که اطلاعاتی را برای جامعه پایتون ارائه می‌دهد، یک ویژگی در جدید پایتون، فرآیندها و یا محیط آن را توصیف می‌کند.

<https://www.python.org/dev/peps/pep-0001>

## ۴-۱. کتابخانه‌ها

در طول این کتاب، به استفاده از کتابخانه‌های موجود در پایتون تمرکز می‌کنیم تا از کد نویسی دوباره اجتناب کنید و با این کار تلاش شما در توسعه برنامه بهینه می‌شود. غالباً به جای نوشتن کد از ابتدایه‌سازکن که فرآیندی زمان‌بر و پرهزینه است، بهتر است برای ایجاد یک شیء، از کلاس‌های از قبل تعریف شده در کتابخانه این زبان استفاده کنید. این کار فقط با یک دستور در پایتون انجام می‌شود؛ بنابراین، کتابخانه‌ها به شما کمک می‌کنند تا با حداقل کد، کارهای قابل توجهی انجام دهید. در این کتاب از طیف گسترده‌ای از کتابخانه‌های استاندارد پایتون، کتابخانه‌های علوم داده و کتابخانه‌های ثالث استفاده خواهید کرد.

### ۴-۱-۱. کتابخانه‌های استاندارد پایتون

کتابخانه استاندارد پایتون قابلیت‌های غنی برای پردازش داده‌های متن / دودویی، ریاضیات، برنامه‌نویسی به سبک تابعی، دسترسی به فایل / دایرکتوری، ماندگاری داده‌ها، فشرده‌سازی / بایگانی داده‌ها، رمزنگاری، سرویس‌های سیستم‌عامل، برنامه‌نویسی هم‌زمانی، ارتباطات بین پردازش‌ها، پروتکل‌های شبکه، XML / JSON و سایر فرمت‌های داده اینترنتی، چندرسانه‌ای، بین‌المللی سازی، رابط کاربری گرافیکی، اشکال‌زدایی، پروفایل و موارد دیگر فراهم می‌کند. در جدول زیر برخی از ماژول‌های کتابخانه استاندارد پایتون ذکر شده است و در مثال‌ها استفاده می‌کنیم.

<sup>۲۴</sup> Zen of Python

<sup>۲۵</sup> Tim Peters

<sup>۲۶</sup> Python Enhancement Proposal

تعدادی از ماژول‌های کتابخانه استاندارد پایتون که در این کتاب استفاده خواهیم کرد.	
ساختارهای داده‌ای اضافی غیر از لیست‌ها، چندتایی‌ها، دیکشنری‌ها و مجموعه‌ها	Collections
پردازش فایل‌هایی که داده‌های آن با کاما جدا شده‌اند.	Csv
کار با تاریخ و زمان	datetime, time
محاسبات ریاضی اعداد صحیح، اعشاری و محاسبات پولی	Decimal
تست واحد ساده از طریق تست‌های اعتبار سنجی و نتایج مورد انتظار تعبیه شده در docstring ها	Doctest
استفاده از JSON برای کار با وب سرویس‌ها و مستندات پایگاه داده‌های نوع NoSQL	Json
ثابت‌های ریاضی متداول و عملگرها	Math
تعامل با سیستم عامل	Os
ساختمان داده صف. اولین داده ورودی اولین داده خروجی است.	Queue
اعداد شبه تصادفی	Random
عبارت‌های با قاعده برای الگوهای ریاضی	Re
دسترسی به پایگاه داده رابطه‌ای با نام SQLite	sqlite3
توابع ریاضی آماری مانند میانگین، مد، میانه و واریانس	Statistics
پردازش رشته	String
پردازش آرگومان‌های خط فرمان، ورودی استاندارد، خروجی استاندارد و جریان خطای استاندارد	Sys
تحلیل کارایی	Timeit

## ۲-۴-۱. کتابخانه‌های علوم داده

پایتون دارای یک انجمن بزرگ و رو به رشد از توسعه‌دهندگان متن‌باز در بسیاری از زمینه‌ها است. کتابخانه‌های متن‌باز که توسط انجمن متن‌باز پایتون توسعه یافته‌اند، یکی از بزرگ‌ترین دلایل محبوبیت این زبان به شمار می‌روند. یکی از اهداف ما ایجاد نمونه‌ها و مطالعات موردی در زمینه پیاده‌سازی است. این کار برای شما یک معرفی جذاب، چالش‌برانگیز و سرگرم‌کننده‌ای در زمینه برنامه‌نویسی پایتون به ارمان می‌آورد و درعین حال شما را با علوم داده، کتابخانه‌های مهم علوم داده و موارد دیگر درگیر می‌کند. از کارهای اساسی که می‌توانید با چند خط کد انجام دهید شگفت‌زده خواهید شد. در جدول زیر کتابخانه‌های رایج علوم داده لیست شده‌اند. بسیاری از این کتابخانه‌ها را هنگام کار با مثال‌های علوم داده مورداستفاده قرار خواهید داد. برای گرافیکی کردن داده‌ها از

## مقدمه‌ای بر کامپیوترها و پایتون ۳۵

Matplotlib، Seaborn و Folium استفاده خواهیم کرد ولی کتابخانه‌های متعددی در این خصوص وجود دارد. برای مشاهده یک لیست خوب و مختصر در مورد کتابخانه‌های گرافیکی پایتون به آدرس <http://pyviz.org/> مراجعه کنید.

<b>کتابخانه‌های معروف پایتون که در علوم داده استفاده می‌شوند.</b>	
<b>محاسبات علمی و آماری</b>	
NumPy	Numerical Python - زبان پایتون فاقد ساختار داده آرایه است. در این زبان از لیست‌ها استفاده می‌شود که البته کار با آن ساده بوده ولی سرعت پایینی دارد. NumPy ساختار داده ndarray با کارایی بالا برای کار با لیست‌ها و ماتریس‌ها ارائه می‌کند. این کتابخانه روتین‌هایی برای پردازش ساختارهای داده‌ای از این دست فراهم می‌کند.
SciPy	SciPy – Scientific Python علاوه بر قابلیت‌های NumPy، روتین‌هایی برای کارهای علمی نظیر انتگرال‌ها، معادلات دیفرانسیل، پردازش ماتریس بیشتر و موارد دیگری را ارائه می‌کند. <a href="http://scipy.org">scipy.org</a> کنترل Python و SciPy را بر عهده دارد
StatsModels	این کتابخانه از تقریب داده‌های آماری، آزمون‌های آماری و تحلیل کاوشگرانه داده‌های آماری پشتیبانی می‌کند.
<b>تحلیل و کار با داده</b>	
pandas	Pandas یک کتابخانه بسیار رایج و محبوب برای کار با داده‌ها است. این کتابخانه به‌طور فزاینده‌ای از ساختار داده ndarray در کتابخانه NumPy استفاده می‌کند. ساختار داده Series (یک‌بعدی) و ساختار داده DataFrame (دو‌بعدی) دو ساختار کلیدی Pandas محسوب می‌شوند.
<b>گرافیک</b>	
Matplotlib	این کتابخانه برای کارهای گرافیکی و رسم نمودار با قابلیت تنظیم و شخصی‌سازی بسیار زیاد است. این کتابخانه از نمودارهای مختلفی نظیر معمولی، پراکندگی، میله‌ای، کاتور، دایروی، شبکه‌ای، محورهای قطبی، سه‌بعدی و متن پشتیبانی می‌کند.
Seaborn	این کتابخانه بر اساس Matplotlib ساخته شده و گرافیک سطح بالاتری ارائه می‌کند. Seaborn کارهای گرافیکی زیباتری ارائه می‌کند و این امکان را به شما می‌دهد تا با کدهای کمتری نمودارهای خود را رسم کنید.
<b>یادگیری ماشینی<sup>۲۷</sup>، یادگیری عمیق<sup>۲۸</sup> و یادگیری تقویتی<sup>۲۹</sup></b>	

این کتابخانه، برترین کتابخانه یادگیری ماشین تلقی می‌شود. یادگیری ماشینی زیرمجموعه هوش مصنوعی است. یادگیری عمیق زیرمجموعه‌ای از یادگیری ماشینی است که بر شبکه‌های عصبی تمرکز دارد.	scikit-learn
از نظر گوگل، TensorFlow پرکاربردترین کتابخانه یادگیری عمیق است. TensorFlow برای کارایی بهتر با gpu (واحدهای پردازش گرافیک) و یا TPU های سفارشی Google (واحدهای پردازش تنسور) کار می‌کند. TensorFlow در هوش مصنوعی و تجزیه و تحلیل کلان داده و در مواردی پردازش‌های حجیم مورد نیاز باشد، اهمیت زیادی دارد. شما از نسخه Keras که بر اساس TensorFlow ساخته شده است، استفاده خواهید.	TensorFlow
OpenAI Gym یک کتابخانه و محیط برای توسعه، مقایسه و آزمون برای الگوریتم‌های یادگیری تقویتی <sup>۳۰</sup> است.	OpenAI Gym
<b>پردازش زبان طبیعی<sup>۳۱</sup> (NLP)</b>	
Natural Language Toolkit - این کتابخانه برای انجام کارهای NLP مورد استفاده قرار می‌گیرد.	NLTK
TextBlob یک کتابخانه پردازش متن NLP با رویکرد شیء‌گرا است که بر اساس NTK و کتابخانه‌های الگوی NLP ساخته شده است. این کتابخانه انجام بسیاری از کارهای NLP را آسان می‌کند.	TextBlob
این کتابخانه شبیه NTK است. معمولاً برای ساخت فهرست یک مجموعه از اسناد استفاده می‌شود و تعیین می‌کند که یک سندی دیگر تا چه حد به اقلام این فهرست شباهت دارد.	Gensim

## ۵-۱. اجرای آزمایشی با استفاده از IPython و Jupyter Notebooks

در این بخش، مفسر IPython را در دو حالت آزمایشی خواهید کرد. قبل از خواندن این بخش، برای نصب توزیع Anaconda Python که شامل مفسر IPython است، دستورالعمل‌های قبل از شروع را دنبال کنید.

<sup>۲۸</sup> Deep Learning

<sup>۲۹</sup> Reinforcement Learning

<sup>۳۰</sup> reinforcement-learning

<sup>۳۱</sup> Natural Language Processing (NLP)

## مقدمه‌ای بر کامپیوترها و پایتون ۳۷

❖ **حالت تعاملی:** در این حالت کدهای کوتاه پایتون به نام **اسنپت**<sup>۳۲</sup> را وارد خواهید کرد و بلافاصله نتایج آن‌ها را مشاهده خواهید کرد.

❖ **حالت اسکریپت:** در این حالت کدهای اجرایی از یک فایل با پسوند PY خوانده می‌شوند و معمولاً این کدها از اسنپت‌ها طولانی‌تر هستند.

در ادامه یاد می‌گیرید چطور از محیط مبتنی بر مرورگر نظیر Jupyter برای نوشتن و اجرای کد پایتون استفاده کنید. Jupyter زبان‌های برنامه‌نویسی زیادی را به شرط آنکه **هسته**<sup>۳۳</sup> آن‌ها را نصب کنید، پشتیبانی می‌کند. برای اطلاعات بیشتر به آدرس ذیل مراجعه کنید:

<https://github.com/jupyter/jupyter/wiki/Jupyter-kernels>

## ۱-۵-۱. استفاده از IPython در وضعیت محاوره‌ای به‌عنوان یک ماشین حساب

بیاید از حالت تعاملی IPython برای ارزیابی عبارات ریاضی ساده استفاده کنیم.

### وارد شدن به IPython در حالت تعاملی

در ابتدا یک پنجره خط فرمان<sup>۳۴</sup> در سیستم‌تان باز کنید:

❖ در سیستم macOS از پوشه Applications و زیرپوشه Utilities آیتم Terminal را باز کنید.

❖ در ویندوز از منوی **شروع**<sup>۳۵</sup> Anaconda Command Prompt را باز کنید.

❖ در لینوکس، Terminal و یا shell سیستم را باز کنید (این مطلب به توزیع لینوکس بستگی دارد).

در خط فرمان ویندوز کلمه ipython را تایپ کرده و سپس Enter کنید. بسته به **پلتفرم**<sup>۳۶</sup> و ورژن پایتون،

پیامی مشابه پیام ذیل را خواهید دید:

```
Python 3.7.0 | packaged by conda-forge | (default, Jan 20 2019, 17:24:52)
Type 'copyright', 'credits' or 'license' for more information
IPython 6.5.0 -- An enhanced Interactive Python. Type '?'
for help.
```

In [1]:

عبارت: In [1] **اعلان**<sup>۳۷</sup> است و نشان می‌دهد که IPython منتظر ورودی شما است. می‌توانید کاراکتر ? را

وارد کنید تا راهنمای سیستم شروع شود و یا آنکه اسنپت‌ها را وارد کنید و البته به‌زودی این کار را خواهید کرد.

<sup>۳۲</sup> snippet

<sup>۳۳</sup> kernel

<sup>۳۴</sup> ommand-line

<sup>۳۵</sup> start menu

<sup>۳۶</sup> platform

<sup>۳۷</sup> prompt

## ارزیابی عبارات ریاضی

در حالت تعاملی می‌توانید عبارات ریاضی را محاسبه کنید:

```
In [1]: 45 + 72
Out[1]: 117
In [2]:
```

بعد از تایپ  $45+72$ ، IPython اسنپت را می‌خواند، آن را ارزیابی کرده و نتیجه در `Out[1]` نشان می‌دهد. در ادامه IPython عبارت `In [2]:` را نشان می‌دهد و معرف آن است که منتظر ورود اسنپت دوم از جانب شما است. IPython برای هر اسنپت به عدد داخل کروشه یک واحد اضافه می‌کند. در این کتاب هر `In [1]` معرف آن است که یک نشست<sup>۳۸</sup> تعاملی جدید آغاز کرده‌اید. این روش را برای هر قسمت یک فصل انجام می‌دهیم. در فصل بعدی خواهید دید که در مواردی عبارت `Out[]` نشان داده نمی‌شود. اجازه دهید عبارتهای پیچیده‌تری را ارزیابی کنیم:

```
In [2]: 5 * (12.7 - 4) / 2
Out[2]: 21.75
```

پایتون از کاراکتر `*` برای عمل ضرب و از کاراکتر `/` برای عمل تقسیم استفاده می‌کند. مشابه ریاضی عبارت داخل پرانتز در ابتدا ارزیابی می‌شوند و به همین دلیل عبارت  $4 - 12.7$  که داخل پرانتز قرار دارد ابتدا محاسبه شده نتیجه  $8.7$  را تولید می‌کند. در ادامه عبارت  $5 * 8.7$  محاسبه شده و نتیجه  $43.5$  حاصل می‌شود. در انتها عبارت  $43.5 / 2$  محاسبه و نتیجه  $21.75$  به دست آمده و IPython آن را در `Out[2]` نشان می‌دهد. اعداد  $4$  و  $2$  اعداد صحیح نامیده می‌شوند. اعدادی نظیر  $12.7$ ،  $43.5$  و  $21.75$  اعداد ممیز شناور نامیده می‌شوند.

## خروج از محیط تعاملی

برای خروج از محیط تعاملی می‌توانید:

❖ در کنار اعلان `In[]` عبارت `exit` را تایپ کنید و با `Enter` کردن فوراً از این محیط خارج شوید.

❖ دکمه `Ctrl + d` را فشار دهید. سیستم عبارت ذیل را نشان می‌دهد:

```
Do you really want to exit ([y]/n?)
```

کروشه‌های دو طرف `y` معرف پاسخ پیش فرض بوده و با `Enter` کردن این پاسخ را تأیید کرده و از محیط خارج می‌شوید.

❖ دکمه `Ctrl + d` را دو بار فشار دهید (فقط در `osMac` و `لینوکس`).

## ۲-۵-۱. اجرای یک برنامه پایتون با استفاده از IPython

در این قسمت یک اسکریپت بنام `RollDieDynamic.py` را اجرا خواهید کرد و در فصل ۶ این برنامه را می‌نویسید. پسوند `py` فایل فوق معرف آن است که این فایل، <sup>۳۹</sup> **کد منبع** یک برنامه پایتون است. اسکریپت `RollDieDynamic.py` پرتاب یک تاس شش‌وجهی را شبیه‌سازی کرده و نمودار تعداد دفعات رخداد هر وجه تاس را به صورت پویا رسم می‌کند. این برنامه این اعداد و نمودار آن‌ها را به صورت رنگی نشان می‌دهد.

### تغییر مسیر کاری به پوشه مثال‌های این فصل

در پوشه `ch01` این کتاب کدهای مربوط به این اسکریپت را پیدا خواهید کرد. قبل از شروع این قسمت پوشه `examples` را در پوشه `Documents حساب کاربری`<sup>۴۰</sup> خود منتقل کنید. هر فصل دارای یک پوشه مربوط به کدهای منبع مربوطه است. هر فصل با فرمت `ch##` نام‌گذاری شده و `##` معرف یک عدد دورقمی از ۰۱ تا ۱۷ می‌باشد. در ابتدا پنجره خط فرمان را باز کنید و سپس با دستور `cd` به مسیر پوشه `ch01` بروید:

❖ در `osMac` و لینوکس دستور `~/Documents/examples/ch0` را تایپ کرده و `Enter` کنید.

❖ در ویندوز دستور ذیل را تایپ کرده و `Enter` کنید.

```
cd C:\Users\YourAccount\Documents\examples\ch01
```

### اجرای اسکریپت

برای اجرای یک اسکریپت دستور ذیل را در خط فرمان تایپ کرده و `Enter` کنید:

```
ipython RollDieDynamic.py 6000 1
```

با این دستور اسکریپت پنجره‌ای باز کرده و نمودار مربوطه را رسم می‌کند. اعداد ۱ و ۶۰۰۰ تعداد تاس‌ها و تعداد پرتاب تاس برای هر تاس را مشخص می‌کنند. در دستور بالا نمودار را ۶۰۰۰ بار برای یک تاس به‌روزرسانی کرده‌ایم.

روی وجوه یک تاس ۶ وجهی اعداد ۱ تا ۶ نقش بسته و احتمال رخداد هر عدد  $\frac{1}{6}$  و یا  $\frac{1}{6}, 667\%$  می‌باشد. اگر تاسی را ۶۰۰۰ بار پرتاب کنیم انتظار داریم هر وجه آن در حدود ۱۰۰۰ بار روی دهد. پرتاب تاس همانند پرتاب یک سکه یک پیشامد کاملاً تصادفی است و به همین دلیل انتظار داریم تعداد دفعات بعضی اعداد کمتر از ۱۰۰۰ و برخی بیشتر از ۱۰۰۰ باشند. در هنگام اجرای اسکریپت از صفحه‌نمایش یک عکس گرفته‌ایم. این اسکریپت از اعدادی که به صورت تصادفی تولید شده‌اند استفاده می‌کند و به همین دلیل نتایج آن با نتایج شما تفاوت دارد. اسکریپت را با تغییر عدد ۱ به ۱۰۰، ۱۰۰۰ و ۱۰۰۰۰ اجرا کنید. زمانی که تعداد دفعات پرتاب

<sup>۳۹</sup> source code

<sup>۴۰</sup> account