

پایتون برای کودکان

مقدمه‌ای سرگرم‌کننده برای برنامه‌نویسی

ویرایش دوم

جیسون. آر. بریگز

مترجم: سید محمد متین روحانی راد

انتشارات پندار پارس

سرشناسه	: بریگز، جیسون آر. Briggs, Jason R.
عنوان و نام پدیدآور	: پایتون برای کودکان: مقدمه‌های سرگرم‌کننده برای برنامه‌نویسی/جیسون آر. بریگز؛ مترجم سید محمد متین‌روحانی‌راد.
مشخصات نشر	: تهران: پندار پارس، ۱۴۰۲.
مشخصات ظاهری	: ۳۰۰ص: مصور.
شابک	: 978-622-7785-25-8
وضعیت فهرست نویسی	: فیپا
یادداشت	: عنوان اصلی: Python for kids : a playful introduction to programming, 2 ed.
عنوان دیگر	: مقدمه‌های سرگرم‌کننده برای برنامه‌نویسی.
موضوع	: پایتون (زبان برنامه‌نویسی کامپیوتر) -- ادبیات کودکان و نوجوانان Python (Computer program language) -- Juvenile literature برنامه‌نویسی -- ادبیات کودکان و نوجوانان Computer programming -- Juvenile literature زبان‌های نوشتاری (کامپیوتر) -- ادبیات کودکان و نوجوانان Scripting languages (Computer science) -- Juvenile literature بازی‌های ویدئویی -- برنامه‌نویسی -- ادبیات کودکان و نوجوانان Video games -- Programming -- Juvenile literature
شناسه افزوده	: روحانی‌راد، سید محمدمتین، ۱۳۷۶-، مترجم
رده بندی کنگره	: QA۷۳/۷۶
رده بندی دیویی	: [ج]۱۳۳/۰۰۵
شماره کتابشناسی ملی	: ۹۴۵۸۱۵۳
اطلاعات رکورد کتابشناسی	: فیپا

انتشارات پندارپارس



دفتر فروش: انقلاب، ابتدای کارگر جنوبی، کوی رشتچی، شماره ۱۴، واحد ۱۶ www.pendarepars.com
تلفن: ۶۶۵۷۲۳۳۵ - ۶۶۹۲۶۵۷۸ همراه: ۰۹۱۲۲۴۵۲۳۴۸ info@pendarepars.com

نام کتاب	: پایتون برای کودکان؛ مقدمه‌ای سرگرم‌کننده برای برنامه‌نویسی (ویرایش دوم)
ناشر	: انتشارات پندار پارس
ترجمه	: سید محمد متین روحانی راد
شمارگان	: ۲۰۰
شابک	: ۹۷۸-۶۲۲-۷۷۸۵-۲۵-۸
چاپ نخست	: دی ماه ۱۴۰۲
	: قیمت
	: ۲۵۰.۰۰۰
	: چاپ، صحافی
	: روز
	: ویراستار
	: حسن یعسوبی
	: جیسون آر. بریگز
	: تالیف
	: ویرایش دوم

*هرگونه کپی برداری، تکثیر و چاپ کاغذی یا الکترونیکی از این کتاب بدون اجازه ناشر تخلف بوده و پیگرد قانونی دارد *

فهرست

مقدمه

۱	چرا پایتون.....
۱	چگونه کدنویسی را بیاموزیم.....
۲	این کتاب مناسب کیست.....
۲	وبسایت کتاب.....
۳	بخش اول: آموزش برنامه‌نویسی.....
۵	فصل ۱: همه مارها نمی‌خزند!.....
۶	چند کلمه در مورد زبان.....
۶	نصب پایتون.....
۷	نصب پایتون روی ویندوز.....
۸	نصب پایتون روی macOS.....
۹	نصب پایتون روی Ubuntu.....
۱۰	نصب پایتون روی سیستم‌عامل Raspberry Pi یا Raspbian.....
۱۲	پس از نصب پایتون.....
۱۳	ذخیره کردن برنامه‌های پایتون.....
۱۵	آنچه در این فصل یاد گرفتید.....
۱۷	فصل ۲: محاسبات و متغیرها.....
۱۷	محاسبه با پایتون.....
۱۹	عملگرهای پایتون.....
۱۹	ترتیب عملیات.....
۲۰	متغیرها مانند برچسب‌ها هستند.....
۲۲	استفاده از متغیرها.....
۲۴	آنچه در این فصل یاد گرفتید.....
۲۵	فصل ۳: رشته‌ها، لیست‌ها، تاپل‌ها و دیکشنری‌ها.....
۲۵	رشته‌ها.....
۲۵	ایجاد رشته‌ها.....
۲۷	رسیدگی به مشکلات با رشته‌ها.....
۲۹	جاسازی مقادیر در رشته‌ها.....
۳۰	ضرب رشته‌ها.....
۳۱	لیست‌ها از رشته‌ها قدرتمندتر هستند.....
۳۳	افزودن موارد به یک لیست.....
۳۴	حذف موارد از یک لیست.....
۳۴	لیست حساب.....
۳۶	تاپل‌ها.....
۳۷	دیکشنری‌های پایتون.....
۳۹	آنچه در این فصل آموختیم.....

۳۹	پازل‌های برنامه‌نویسی
۳۹	شماره ۱: موارد دلخواه
۴۰	شماره ۲: شمارش رزمندگان
۴۰	شماره ۳: خوش‌آمدگویی!
۴۰	شماره ۴: نامه چند خطی
۴۱	فصل ۴: طراحی با لاک‌پشت
۴۱	استفاده از ماژول TURTLES در پایتون
۴۲	ایجاد یک بوم
۴۳	حرکت لاک‌پشت
۴۸	آنچه در این فصل یاد گرفتید
۴۸	پازل‌های برنامه‌نویسی
۴۸	شماره ۱: یک مستطیل
۴۸	شماره ۲: مثلث
۴۸	شماره ۳: یک جعبه بدون گوشه
۴۹	شماره ۴: یک جعبه کج شده بدون گوشه
۵۱	فصل ۵: پرسیدن سؤال با IF و ELSE
۵۱	عبارت‌های if
۵۲	بلوک، گروهی از عبارات برنامه‌نویسی است
۵۴	شرط‌ها به ما کمک می‌کنند تا چیزها را با هم مقایسه کنیم
۵۵	عبارت‌های if-THEN-ELSE
۵۶	عبارت‌های if و ELIF
۵۷	شرط‌های ترکیبی
۵۸	متغیرهای بدون مقدار NONE
۵۹	تفاوت بین رشته‌ها و اعداد
۶۱	آنچه در این فصل یاد گرفتید
۶۲	پازل‌های برنامه‌نویسی
۶۲	شماره ۱: آیا شما ثروتمند هستید؟
۶۲	شماره ۲: چشمک‌زن! TWINKIES!
۶۲	شماره ۳: فقط عدد مناسب
۶۲	شماره ۴: من می‌توانم با آن نینجاها مبارزه کنم
۶۳	فصل ۶: حلقه زدن (LOOPING)
۶۳	استفاده از حلقه FOR
۷۰	وقتی در مورد حلقه زدن صحبت می‌کنیم
۷۳	آنچه در این فصل یاد گرفتید
۷۳	پازل‌های برنامه‌نویسی
۷۳	شماره ۱: حلقه سلام
۷۳	شماره ۲: اعداد زوج
۷۴	شماره ۳: پنج ماده مورد علاقه من

۷۴.....	شماره ۴: وزن شما در ماه.....
۷۵.....	فصل ۷: بازیافت کد با توابع و ماژول‌ها.....
۷۶.....	استفاده از توابع.....
۷۶.....	بخش‌هایی از یک تابع.....
۷۷.....	متغیرها و قلمرو.....
۸۰.....	استفاده از ماژول‌ها.....
۸۲.....	تابع ورودی.....
۸۳.....	آنچه در این فصل یاد گرفتید.....
۸۳.....	پازل‌های برنامه‌نویسی.....
۸۳.....	شماره ۱: تابع پایه وزن ماهانه.....
۸۴.....	شماره ۲: تابع وزن ماهانه و سالانه.....
۸۴.....	شماره ۳: برنامه وزن ماهانه.....
۸۴.....	شماره ۴: برنامه وزن در مریخ.....
۸۵.....	فصل ۸: نحوه استفاده از کلاس‌ها و اشیاء.....
۸۶.....	تقسیم کردن چیزها به کلاس‌ها.....
۸۷.....	فرزندان و والدین.....
۸۸.....	افزودن اشیاء به کلاس‌ها.....
۸۸.....	تعریف توابع کلاس‌ها.....
۸۹.....	اضافه کردن مشخصات کلاس.....
۹۱.....	چرا از کلاس‌ها و اشیاء استفاده کنیم؟.....
۹۲.....	اشیاء و کلاس‌ها در تصاویر.....
۹۴.....	سایر ویژگی‌های مفید اشیاء و کلاس‌ها.....
۹۵.....	توابع ارثی.....
۹۶.....	توابع، توابع دیگر را فراخوانی می‌کنند.....
۹۸.....	مقداردهی اولیه به یک شیء.....
۹۹.....	آنچه در این فصل یاد گرفتید.....
۹۹.....	پازل‌های برنامه‌نویسی.....
۹۹.....	شماره ۱: اختلاط زرافه.....
۱۰۰.....	شماره ۲: پنجه لاک‌پشت.....
۱۰۰.....	شماره ۳: دو مارپیچ کوچک.....
۱۰۰.....	شماره ۴: چهار مارپیچ کوچک.....
۱۰۱.....	فصل ۹: گرافیک بیشتر لاک‌پشت.....
۱۰۱.....	با مربع اصلی شروع کنید.....
۱۰۲.....	طراحی ستاره‌ها.....
۱۰۶.....	طراحی ماشین.....
۱۰۸.....	رنگ‌آمیزی چیزها.....
۱۰۸.....	تابعی برای رسم یک دایره پر شده.....
۱۰۹.....	ایجاد سیاه و سفید خالص.....

۱۱۰	تابع رسم مربع
۱۱۱	طراحی مربع‌های پر شده
۱۱۳	طراحی ستاره‌های پر شده
۱۱۵	آنچه در این فصل یاد گرفتید
۱۱۵	پازل‌های برنامه‌نویسی
۱۱۵	شماره ۱: رسم یک هشت ضلعی
۱۱۶	شماره ۲: رسم یک هشت ضلعی پر شده
۱۱۶	شماره ۳: یکی دیگر از توابع رسم ستاره
۱۱۶	شماره ۴: چهار مارپیچ بازبینی شده است
۱۱۷	فصل ۱۰: استفاده از TKINTER برای گرافیک بهتر
۱۱۸	ایجاد یک دکمه قابل کلیک
۱۲۰	استفاده از پارامترهای نام‌گذاری شده
۱۲۱	ایجاد یک بوم برای طراحی
۱۲۲	طراحی خطوط
۱۲۳	طراحی جعبه‌ها
۱۲۵	رسم تعداد زیادی مستطیل
۱۲۷	تنظیم رنگ
۱۳۰	طراحی قوس
۱۳۲	رسم چند ضلعی
۱۳۳	نمایش متن
۱۳۵	نمایش تصاویر
۱۳۷	ایجاد انیمیشن پایه
۱۳۹	واکنش یک شیء به چیزی
۱۴۲	روش‌های بیشتر برای استفاده از شناسه
۱۴۳	آنچه در این فصل یاد گرفتید
۱۴۳	پازل‌های برنامه‌نویسی
۱۴۳	شماره ۱: صفحه را با مثلث‌ها پر کنید
۱۴۴	شماره ۲: مثلث متحرک
۱۴۴	شماره ۳: عکس متحرک
۱۴۴	شماره ۴: صفحه را با عکس پر کنید
۱۴۵	بخش دوم: BOUNCE!
۱۴۷	فصل ۱۱: شروع اولین بازی شما: BOUNCE!
۱۴۷	ضربه زدن به توپ پرنده
۱۴۸	ایجاد بوم بازی
۱۴۹	ایجاد کلاس BALL
۱۵۱	افزودن برخی اقدامات
۱۵۲	حرکت دادن توپ
۱۵۳	جهش توپ

۱۵۵	تغییر جهت شروع حرکت توپ
۱۵۷	آنچه در این فصل یاد گرفتید
۱۵۷	پازل‌های برنامه‌نویسی
۱۵۷	شماره ۱: تغییر رنگ‌ها
۱۵۷	شماره ۲: رنگ‌های چشمک زن
۱۵۷	شماره ۳: مواضع خود را بگیرید!
۱۵۸	شماره ۴: اضافه کردن پارو. . . ؟
۱۵۹	فصل ۱۲: پایان اولین بازی: BOUNCE!
۱۵۹	اضافه کردن پارو
۱۶۱	حرکت دادن پارو
۱۶۲	پیدا کردن زمانی که توپ به پارو برخورد می‌کند
۱۶۵	اضافه کردن یک عنصر شانس
۱۶۸	آنچه در این فصل یاد گرفتید
۱۶۸	پازل‌های برنامه‌نویسی
۱۶۸	شماره ۱: شروع بازی را به تأخیر بیندازید
۱۶۸	شماره ۲: یک "GAME OVER" مناسب
۱۶۹	شماره ۳: به توپ شتاب دهید
۱۶۹	شماره ۴: امتیاز بازیکن را ثبت کنید
۱۷۱	بخش سوم: آقای استیکمن برای خروج مسابقه می‌دهد
۱۷۳	فصل ۱۳: ایجاد گرافیک برای بازی MR. STICK MAN
۱۷۳	طرح بازی MR. STICK MAN
۱۷۴	دریافت GIMP
۱۷۶	ایجاد عناصر بازی
۱۷۶	تهیه یک تصویر شفاف
۱۷۷	طراحی آقای استیکمن
۱۷۸	آقای استیکمن که در حال دویدن به سمت راست است
۱۷۹	آقای استیکمن که در حال دویدن به سمت چپ است
۱۷۹	طراحی سکوها
۱۸۰	طراحی درب
۱۸۱	طراحی پس‌زمینه
۱۸۲	شفافیت
۱۸۳	آنچه در این فصل یاد گرفتید
۱۸۵	فصل ۱۴: توسعه بازی MR. STICK MAN
۱۸۵	ایجاد کلاس GAME
۱۸۵	تنظیم عنوان پنجره و ایجاد بوم
۱۸۶	تکمیل تابع __INIT__
۱۸۸	ایجاد تابع MAINLOOP
۱۹۰	ایجاد کلاس COORDS

۱۹۰	بررسی برخورد
۱۹۱	برخورد اسپرایت‌ها به صورت افقی
۱۹۳	برخورد اسپرایت‌ها به صورت عمودی
۱۹۳	قرار دادن همه اینها با هم: کد تشخیص برخورد نهایی
۱۹۳	تابع COLLIDED_LEFT
۱۹۴	تابع COLLIDED_RIGHT
۱۹۵	تابع COLLIDED_TOP
۱۹۵	تابع COLLIDED_BOTTOM
۱۹۶	ایجاد کلاس SPRITE
۱۹۷	افزودن سکوها
۱۹۸	افزودن یک شیء سکو
۱۹۹	افزودن دسته‌ای از سکوها
۲۰۰	آنچه در این فصل یاد گرفتید
۲۰۱	پازل‌های برنامه‌نویسی
۲۰۱	شماره ۱: صفحه شطرنجی
۲۰۱	شماره ۲: صفحه شطرنجی با دو تصویر
۲۰۱	شماره ۳: قفسه کتاب و چراغ
۲۰۲	شماره ۴: پس‌زمینه تصادفی
۲۰۳	فصل ۱۵: ایجاد آقای استیکمن
۲۰۳	اولیه‌سازی شکل چوبی
۲۰۴	بارگذاری تصاویر شکل چوبی
۲۰۵	راه‌اندازی متغیرها
۲۰۶	اتصال به کلیدها
۲۰۶	چرخاندن شکل چوبی به چپ و راست
۲۰۷	ساختن JUMP شکل چوبی
۲۰۸	آنچه تا این‌جا کار داریم
۲۱۰	آنچه در این فصل یاد گرفتید
۲۱۱	فصل ۱۶: تکمیل بازی آقای استیکمن
۲۱۱	متحرک‌سازی شکل چوبی
۲۱۲	ایجاد تابع ANIMATE
۲۱۲	بررسی حرکت
۲۱۴	تغییر تصویر
۲۱۵	گرفتن موقعیت شکل چوبی
۲۱۶	حرکت دادن شکل چوبی
۲۱۶	شروع تابع MOVE
۲۱۸	آیا شکل چوبی به پایین یا بالای بوم برخورد کرده است؟
۲۱۸	آیا شکل چوبی به دو طرف بوم برخورد کرده است؟
۲۱۹	برخورد با اسپرایت‌های دیگر

۲۲۰	برخورد در پایین
۲۲۱	بررسی چپ و راست
۲۲۳	آزمایش اسپرایت شکل چوبی
۲۲۴	خروج!
۲۲۴	ایجاد کلاس DOORSPRITE
۲۲۵	تشخیص درب
۲۲۵	افزودن شیء DOOR
۲۲۶	بازی نهایی
۲۳۲	آنچه در این فصل یاد گرفتید
۲۳۲	پازل‌های برنامه‌نویسی
۲۳۲	شماره ۱: "YOU WIN!"
۲۳۲	شماره ۲: متحرک کردن درب
۲۳۳	شماره ۳: سکوهاى متحرک
۲۳۳	شماره ۴: لامپ به عنوان اسپرایت
۲۳۵	فصل ۱۷: گام بعدی چیست؟
۲۳۶	نصب پایتون PIP روی ویندوز
۲۳۶	نصب پایتون PIP روی اوبونتو
۲۳۷	نصب پایتون PIP روی رزبری‌پای
۲۳۸	نصب پایتون PIP در MACOS
۲۳۸	آزمایش PYGAME
۲۴۰	سایر بازی‌ها و گرافیک‌های برنامه‌نویسی
۲۴۰	سایر زبان‌های برنامه‌نویسی
۲۴۱	JAVASCRIPT
۲۴۱	JAVA
۲۴۱	C#
۲۴۲	C/C++
۲۴۳	RUBY
۲۴۳	GO
۲۴۳	RUST
۲۴۴	SWIFT
۲۴۴	کلمات پایانی
۲۴۵	پیوست الف؛ کلیدواژه‌های پایتون
۲۴۵	AND
۲۴۶	AS
۲۴۶	ASSERT
۲۴۷	ASYNC
۲۴۷	AWAIT
۲۴۷	BREAK

۲۴۸.....	CLASS
۲۴۸.....	CONTINUE
۲۴۹.....	DEF
۲۴۹.....	DEL
۲۴۹.....	ELIF
۲۵۰.....	ELSE
۲۵۰.....	EXCEPT
۲۵۰.....	FINALLY
۲۵۰.....	FOR
۲۵۰.....	FROM
۲۵۲.....	GLOBAL
۲۵۲.....	IF
۲۵۲.....	IMPORT
۲۵۲.....	IN
۲۵۳.....	IS
۲۵۴.....	LAMBDA
۲۵۴.....	NONLOCAL
۲۵۴.....	NOT
۲۵۴.....	OR
۲۵۵.....	PASS
۲۵۶.....	RAISE
۲۵۷.....	RETURN
۲۵۷.....	TRY
۲۵۷.....	WHILE
۲۵۸.....	WITH
۲۵۸.....	YIELD
۲۵۹.....	پیوست ب؛ توابع داخلی پایتون
۲۵۹.....	استفاده از توابع داخلی
۲۵۹.....	تابع ABS
۲۶۱.....	تابع ALL
۲۶۱.....	تابع ANY
۲۶۲.....	تابع BIN
۲۶۲.....	تابع BOOL
۲۶۴.....	تابع CALLABLE
۲۶۴.....	تابع CHR
۲۶۵.....	تابع DIR
۲۶۶.....	تابع DIVMOD
۲۶۷.....	تابع EVAL

۲۶۸.....	تابع EXEC
۲۶۹.....	تابع FLOAT
۲۷۰.....	تابع INPUT
۲۷۰.....	تابع INT
۲۷۱.....	تابع LEN
۲۷۲.....	تابع LIST
۲۷۲.....	توابع MAX و MIN
۲۷۴.....	تابع ORD
۲۷۴.....	تابع POW
۲۷۵.....	تابع RANGE
۲۷۶.....	تابع SUM
۲۷۶.....	باز کردن یک فایل در پایتون
۲۷۶.....	باز کردن یک فایل ویندوز
۲۷۷.....	باز کردن یک فایل MACOS
۲۷۷.....	باز کردن یک فایل اوبونتو یا رزبری پای
۲۷۸.....	نوشتن در فایل‌ها
۲۷۹.....	پیوست پ؛ عیب‌یابی
۲۷۹.....	خطاهای "TK" در وارد کردن لاک‌پشت در اوبونتو
۲۷۹.....	خطای ATTRIBUTE در استفاده از لاک‌پشت
۲۸۰.....	مشکلات اجرای لاک‌پشت
۲۸۱.....	کلاس بدون آرگومان‌ها

سخن ناشر

ستایش خدای را که به بندگانش، موهبت عقل، اندیشه و تفکر را ارزانی داشت. ما انسان‌ها موجوداتی پر پیچ و خم، خلاق، جسور، شگفت‌انگیز، و جذاب هستیم؛ و با وجودی که در گستره هستی، ذره‌ای بیش نیستیم، اما با قدرت‌های فراوان و ناشناخته‌ای که داریم، قادر به تسخیر کهکشان‌ها، گذر از ابعاد زمان و مکان، و دستیابی به هر ناشناخته و مجهولی می‌باشیم.

علوم کامپیوتری و زبان‌های برنامه‌نویسی، ابزارهایی است که ذره‌ای از قدرت فکر انسان را بروز می‌دهد و تجلی‌گر پیچیدگی‌های ذهنی انسان‌های تلاش‌گر است. نسل جدید، نسل ارتباطات و فنآوری است. خوراکش اینترنت، فضای مجازی، و دیتاست. باید به فکر این نسل بود و خوراک سالم در اختیارش گذاشت. در آینده‌ای نه چندان دور، به‌کمک هوش‌های مصنوعی، وارد دنیاهای مجازی و واقعی دیگری خواهیم شد و این نسل، نباید از آنها عقب بماند.

نسخه زبان اصلی این کتاب، که بیش از ۲۵۰ هزار کپی از آن به فروش رفته، به زبانی ساده تلاش می‌کند با خلق بازی‌های کامپیوتری، شما را با زبان جذاب پایتون آشنا کند. به آن اکتفا نکنید و با بسط مثال‌ها و تمرین‌ها، خلاقیت خود را بیازمایید. پس از یادگیری خوب مبانی پایتون، کتاب آموزش برنامه‌نویسی Python 3.9 پندار پارس، منتظر شماست تا شما را به یک برنامه‌نویس حرفه‌ای تبدیل کند. به امید آن روز...

مفتخریم حدود ۲۰ سال در این حوزه فعالیم و آمده‌ایم که بمانیم و در این راه خدمت کنیم. به امید روزی که شما هم یکی از نویسندگان و مترجمان ما باشید. پس همراه ما باشید.

مهندس حسین یعسوبی

مدیر مسئول انتشارات پندار پارس

مقدمه

چرا برنامه‌نویسی کامپیوتر را یاد بگیریم؟ برنامه‌نویسی، خلاقیت، استدلال و حل مسئله را تقویت می‌کند. برنامه‌نویس، این فرصت را پیدا می‌کند که چیزی از هیچ بسازد، از منطق استفاده کند تا ساختارهای برنامه‌نویسی را به شکلی تبدیل کند که یک کامپیوتر بتواند اجرا کند، و زمانی که همه چیز آن‌طور که انتظار می‌رود کار نمی‌کند، از حل مسئله استفاده می‌کند تا بفهمد چه چیزی اشتباه شده است. برنامه‌نویسی یک فعالیت سرگرم‌کننده، گاهی چالش‌برانگیز (و گاهی خسته‌کننده) است، و مهارت‌های آموخته شده از آن می‌تواند هم در مدرسه و هم در محل کار مفید باشد - حتی اگر حرفه شما ربطی به کامپیوتر نداشته باشد. و اگر هیچ چیز دیگری نباشد، برنامه‌نویسی یک راه عالی برای گذراندن یک بعد از ظهر است که هوای بیرون دلگیر است.

چرا پایتون

افزون بر سادگی ساختار زبانی آن و یک پوسته تعاملی برای آزمایش، پایتون دارای ویژگی‌هایی است که روند یادگیری را تا حد زیادی تقویت می‌کند و به شما امکان می‌دهد انیمیشن‌های ساده‌ای را برای ایجاد بازی‌های خود کنار هم قرار دهید. یکی از آنها ماژول turtle است، که از گرافیک (لاک‌پشت) الهام گرفته شده است (که در دهه ۱۹۶۰ توسط زبان برنامه‌نویسی Logo استفاده می‌شد) و برای استفاده آموزشی طراحی شده بود. ماژول دیگر، tkinter نام دارد که یک واسط برای تولکیت رابط کاربری گرافیکی TK است، که یک روش ساده برای ایجاد برنامه‌ها با گرافیک‌های کاملاً پیشرفته‌تر و انیمیشنی است.

چگونه کدنویسی را بیاموزیم

مانند هرکاری، نیاز به تلاش ابتدایی دارد و همیشه بهتر است با مبانی ابتدایی کار کنید، پس با فصل‌های آغازین کتاب، شروع به تمرین کنید و به آرامی به فصل‌های بعدی بروید. هیچ کس نمی‌تواند یک سمفونی ارکسترال را نخستین باری که ساز را به دست می‌گیرد بنوازد. دانش‌جویان خلبانی پیش از اینکه کنترل‌های اولیه را درک کنند، پرواز با هواپیما را شروع نمی‌کنند. ژیمناست‌ها (معمولاً) نمی‌توانند در نخستین تلاش، خود را به عقب پرتاب کنند. اگر خیلی سریع پیش بروید، نه تنها ایده‌های اساسی در ذهن شما باقی نمی‌ماند، بلکه محتوای فصل‌های بعدی را پیچیده‌تر از آنچه در واقعیت است پیدا خواهید کرد.

همان‌گونه که در کتاب جلو می‌روید، سعی کنید ببینید مثال‌ها چگونه کار می‌کند. بیشتر فصل‌ها نیز شامل پازل‌های برنامه‌نویسی است که با امتحان کردن آنها بتوانید با مبانی کار، بهتر آشنا شوید و به شناخت بهتر ایده‌های پیچیده‌تر کمک می‌کند. اگر در درک موضوعی نا امید شدید یا چالش زیادی داشتید، نکات زیر می‌تواند به کمک شما بیاید:

۱. مشکل را به بخش‌های کوچک‌تری تقسیم کنید. سعی کنید بفهمید یک قطعه کوچک کد چه کاری انجام می‌دهد یا فقط به بخش کوچکی از یک ایده دشوار فکر کنید (به جای تلاش برای درک یک‌باره همه چیز، روی یک قطعه کوچک از کد تمرکز کنید).
۲. اگر این کمک نکرد، بهتر است آنرا رها کنید و روز دیگری به سراغش بیایید. این روش خوبی برای حل بسیاری از مشکلات روزمره زندگی است و برای برنامه‌نویسی هم کارساز است.

این کتاب مناسب کیست

این کتاب، مناسب هر فردی است که علاقه‌مند به برنامه‌نویسی است، چه کودک باشد چه بزرگ‌سالی که برای نخستین بار، وارد برنامه‌نویسی می‌شود. اگر می‌خواهید به‌جای استفاده از نرم‌افزارهایی که دیگران توسعه داده‌اند، بیاموزید چگونه نرم‌افزار شخصی خودتان را بنویسید، این کتاب، نقطه آغاز مناسبی برای شماست.

وبسایت کتاب

همه مثال‌های کتاب را می‌توانید از وبسایت کتاب به نشانی <http://python-for-kids.com> دانلود کنید. هرچند، در سایت انتشارات پندار پارس هم در صفحه این کتاب، قابل دانلود است.

بخش اول

آموزش برنامه نویسی

فصل ۱

همه مارها نمی خزند!



یک برنامه کامپیوتر، مجموعه‌ای از دستورالعمل‌ها است که باعث می‌شود کامپیوتر کاری انجام دهد. این بخش‌های فیزیکی رایانه نیست (مانند سیم‌ها، ریزتراشه‌ها، کارت‌ها، هارد دیسک و مواردی از این دست)، بلکه چیزهای پنهانی هستند که روی آن سخت‌افزار اجرا می‌شوند. یک برنامه کامپیوتر، که من معمولاً به آن فقط به عنوان برنامه اشاره می‌کنم، مجموعه‌ای از دستورالعمل‌ها است که به این سیستم می‌گوید چه کاری باید انجام دهد. نرم‌افزار مجموعه‌ای از برنامه‌های کامپیوتری است.

تقریباً تمامی وسایل روزانه بدون برنامه‌های کامپیوتری، کارایی خود را از دست می‌دهند و یا بسیار کمتر مفید هستند. برنامه‌های کامپیوتری نه تنها کامپیوتر شخصی شما را به شکلی مختلف کنترل می‌کنند، بلکه سیستم‌های ویدئویی، تلفن‌های همراه و سیستم‌های جی پی اس در ماشین‌ها را هم کنترل می‌کنند. برنامه‌ها همچنین برخی از کالاها مثل تلویزیون‌های LCD و کنترل از راه دور آنها، رادیوهای جدید، پخش کننده دی وی دی، فرستنده‌های الکترونیکی و برخی از یخچال‌ها را کنترل می‌کنند. حتی موتورهای ماشین، چراغ‌های خیابان، علامت‌های قطار، صفحه‌های الکترونیکی آگهی‌ها و آسانسورها توسط برنامه‌ها کنترل می‌شوند.

برنامه‌ها کمی شبیه افکار هستند. اگر افکاری نداشتید، احتمالاً روی زمین می‌نشستید و به دیواری خیره می‌شدید. فکر "از زمین بلند شو" دستورالعمل یا فرمانی است که به بدن شما می‌گوید بلند شود. به همین ترتیب برنامه‌های کامپیوتری از دستورالعمل‌ها استفاده می‌کنند تا به کامپیوترها بگویند چه کاری انجام دهند.

اگر بدانید چگونه برنامه کامپیوتری بنویسید می‌توانید کارهای مفیدی انجام دهید. البته ممکن است نتوانید برنامه‌هایی برای کنترل ترافیک، چراغ‌های راهنما، یا یخچال‌تان بنویسید (خب، حداقل در ابتدا نه)؛ اما می‌توانید صفحات وب ایجاد کنید، بازی‌های خودتان را بنویسید، یا حتی برنامه‌ای برای کمک به تکالیف‌تان بسازید.

چند کلمه در مورد زبان

کامپیوترها هم مانند انسان‌ها، از چندین زبان برای ارتباط استفاده می‌کنند. این زبان‌ها به زبان‌های برنامه‌نویسی معروف هستند. زبان برنامه‌نویسی، راه ساده‌ای برای صحبت با کامپیوتر با استفاده از دستورالعمل‌هایی است که هم انسان و هم کامپیوتر می‌توانند آن را درک کنند.

زبان‌های برنامه‌نویسی به نام آدم‌ها نام‌گذاری شده‌اند (مانند Ada و Pascal)، زبان‌هایی هم با استفاده از ترکیبات ساده نام‌گذاری شده‌اند (مانند BASIC و FORTRAN)، حتی تعدادی از آنها به نام سریال‌های تلویزیونی مانند Python نام‌گذاری شده‌اند. بله، نام زبان برنامه‌نویسی پایتون از روی نمایش تلویزیونی Monty Python's Flying Circus گرفته شده است، نه از مار.

توجه: نمایش Monty Python's Flying Circus یک کمدی تلویزیونی بریتانیا بود که در سال‌های دهه ۱۹۷۰ پخش شد و هنوز هم در میان تماشاگران خاصی بسیار محبوب است. طرح‌هایی از قبیل «وزارت گردش‌های احمقانه»، «رقص ماهی» و «فروشگاه پنیری» (که پنیری نمی‌فروخت) در این نمایش دیده می‌شد.

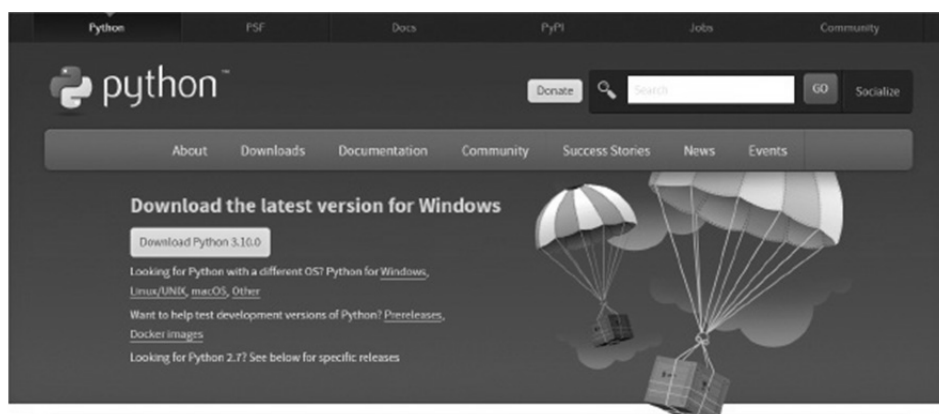
زبان برنامه‌نویسی پایتون امکانات بسیاری دارد که آن را برای مبتدیان بسیار مفید می‌کند. مهم‌تر از همه اینکه شما می‌توانید با پایتون برنامه‌های ساده و کارآمد را خیلی سریع بنویسید. پایتون به اندازه زبان‌های دیگر، از نمادهای پیچیده استفاده نمی‌کند که این موجب شده خواندن آن آسان‌تر و برای مبتدیان دوست‌داشتنی‌تر شود. (منظورمان این نیست که پایتون از نمادها استفاده نمی‌کند. آنها به اندازه بسیاری از زبان‌های دیگر به شدت استفاده نمی‌شوند.)

نصب پایتون

نصب پایتون نسبتاً ساده است. در اینجا، ما مراحل مربوط به نصب آن روی ویندوز، macOS، Ubuntu و Raspberry Pi را مرور می‌کنیم. هنگام نصب پایتون، برنامه IDLE را نیز نصب خواهید کرد، که محیط توسعه یکپارچه‌ای است که به شما این امکان را می‌دهد با پایتون برنامه بنویسید. اگر پایتون قبلاً روی کامپیوتر شما نصب شده است، به بخش "پس از نصب پایتون" مراجعه کنید.

نصب پایتون روی ویندوز

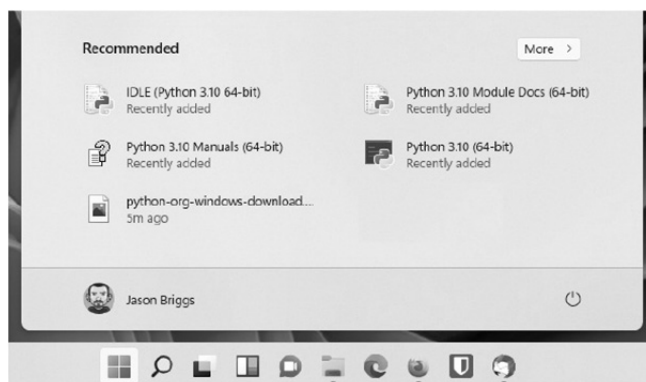
برای نصب پایتون روی مایکروسافت ویندوز ۱۱، نسخه Python 3.10 یا بالاتر را از <http://www.python.org/downloads/> دانلود کنید. نسخه دقیق پایتون که دانلود می‌کنید تا زمانی که حداقل نسخه ۳.۱۰ باشد، اهمیتی ندارد. اما اگر از نسخه قدیمی ویندوز استفاده می‌کنید (مانند ویندوز ۷)، آخرین نسخه پایتون کار نخواهد کرد. در این مورد، باید پایتون ۳.۸ را نصب کنید. در مورد اینکه کدام نسخه از پایتون با نسخه ویندوز شما کار خواهد کرد، به صفحه دانلود ویندوز مراجعه کنید (<https://www.python.org/downloads/windows>).



شکل ۱-۱: دانلود پایتون برای ویندوز

اگر مرورگر شما می‌پرسد که آیا فایل شما ذخیره شود یا باز شود، ذخیره شود را انتخاب کنید. پس از دانلود فایل نصب پایتون برای ویندوز، باید اقدام به اجرای آن کنید. در غیر این صورت، پوشه Downloads را باز کنید و روی فایل اجرایی، دوبار کلیک کنید. اکنون، دستورالعمل‌های نصب روی صفحه را دنبال کنید و پایتون را در محل پیش فرض به صورت زیر نصب کنید:

۱. روی **Install Now** کلیک کنید.
۲. وقتی از شما پرسیده شد که آیا به برنامه اجازه می‌دهید تغییراتی در دستگاه شما ایجاد کند، **Yes** را انتخاب کنید.
۳. پس از اتمام نصب، روی **Close** کلیک کنید و باید تعدادی آیکن Python 3.1x را در منوی استارت ویندوز خود مشاهده کنید:



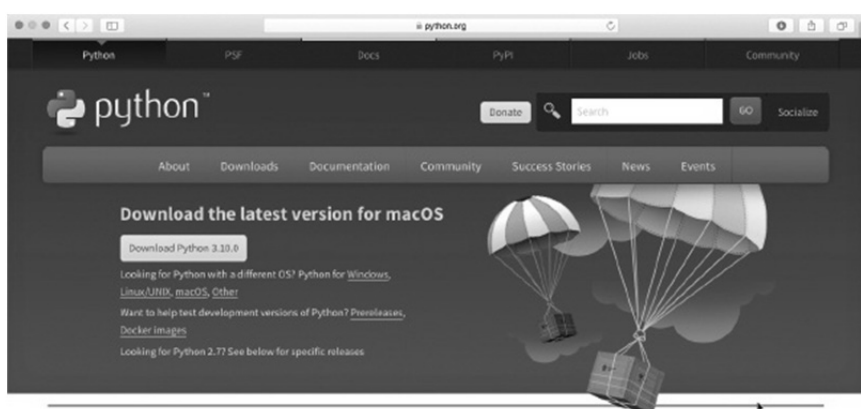
شکل ۱-۲: منوی استارت ممکن است با توجه به نسخه پایتونی که استفاده می‌کنید متفاوت باشد.

اینک برای شروع کار با پایتون به بخش "پس از نصب پایتون" مراجعه کنید.

نصب پایتون روی macOS

اگر از مک استفاده می‌کنید باید نسخه‌ای از پایتون را که پیش‌نصب شده پیدا کنید، اما این نسخه ممکن است نسخه قدیمی این زبان باشد. برای اطمینان از این که نسخه جدیدی را اجرا می‌کنید، باید روی آیکن نورافکن (ذره‌بین در گوشه بالای سمت راست)، کلیک کنید و terminal را در پنجره ظاهر شده تایپ کنید. هنگامی که terminal باز شد، `python3 --version` را تایپ کنید (دو خط فاصله و به دنبال آن کلمه version) و ENTER را بزنید.

اگر می‌بینید که دستور یافت نشد یا نسخه‌ای کمتر از ۳.۱۰ است، برای دانلود آخرین ورژن اپلیکیشن برای macOS به آدرس زیر بروید: <http://www.python.org/downloads/>.



شکل ۱-۳: دانلود پایتون برای macOS

پس از دانلود، روی فایل دوبار کلیک کنید (باید چیزی مانند python-3.10.0-macosx11.pkg باشد) با مجوز موافقت کنید و دستورالعمل‌های روی صفحه را برای نصب نرم‌افزار دنبال کنید. پیش از نصب پایتون، باید رمز عبور مدیر سیستم مک از شما خواسته شود. اگر رمز عبور را ندارید، از والدین خود یا هر کسی که صاحب دستگاه شما است بپرسید.



شکل ۱-۴: پایتون در Mac Finder

برای شروع کار با پایتون به "هنگامی که پایتون را نصب کردید" مراجعه کنید.

نصب پایتون روی Ubuntu

پایتون روی لینوکس اوبونتو از پیش نصب شده است، اما ممکن است جدیدترین نسخه نباشد. برای دریافت آخرین نسخه پایتون، این دستورالعمل‌ها را دنبال کنید (توجه داشته باشید که ممکن است لازم باشد شماره نسخه را در دستور زیر تغییر دهید تا آخرین نسخه منعکس شود):

۱. روی نماد Show Applications (معمولاً ۹ نقطه در گوشه سمت چپ پایین صفحه) کلیک کنید.
۲. terminal را در کادر ورودی وارد کنید (یا اگر از پیش نمایش داده شده است، روی Terminal کلیک کنید).
۳. وقتی پنجره terminal باز شد، عبارت زیر را وارد کنید:

```
sudo apt update
sudo apt install python3.10 idle-python3.10
```

ممکن است پس از وارد کردن اولین دستور از شما خواسته شود که رمز عبور مدیر سیستم را وارد کنید (اگر رمز عبور مدیر سیستم را ندارید، لازم است از والدین یا معلم بخواهید آن را وارد کنند).

```

jasonbriggs@jasonbriggs-VirtualBox: ~
jasonbriggs@jasonbriggs-VirtualBox: ~$ sudo apt install python3.10 idle-python3.10
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  blt fonts-mathjax libjs-mathjax libpython3.10-minimal libpython3.10-stdlib
  libtk8.6 python3-tk python3.10-minimal tk8.6-blt2.5
Suggested packages:
  blt-demo fonts-mathjax-extras fonts-stix libjs-mathjax-doc tk8.6 tix
  python3-tk-dbg python3.10-venv python3.10-doc binfmt-support
The following NEW packages will be installed:
  blt fonts-mathjax idle-python3.10 libjs-mathjax libpython3.10-minimal
  libpython3.10-stdlib libtk8.6 python3-tk python3.10 python3.10-minimal
  tk8.6-blt2.5
0 to upgrade, 11 to newly install, 0 to remove and 0 not to upgrade.
Need to get 15.0 MB of archives.
After this operation, 73.3 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://gb.archive.ubuntu.com/ubuntu/impish/universe amd64 libpython3.10-minimal amd64 3.10.0-2 [802 kB]
Get:2 http://gb.archive.ubuntu.com/ubuntu/impish/universe amd64 python3.10-minimal amd64 3.10.0-2 [2,115 kB]
Get:3 http://gb.archive.ubuntu.com/ubuntu/impish/main amd64 libtk8.6 amd64 8.6.1

```

شکل ۱-۵: نصب پایتون در ترمینال در اوبونتو. با توجه به نسخه‌ای که دانلود می‌کنید ممکن است خروجی شما کمی متفاوت به نظر برسد.

برای شروع کار با پایتون به بخش "پس از نصب پایتون" مراجعه کنید.

نصب پایتون روی سیستم‌عامل Raspberry Pi یا Raspbian

پایتون ۳ در سیستم عامل رزبری‌پای از پیش نصب شده است، اما در حال حاضر، نسخه ۳.۷ است. نصب نسخه بعدی کمی پیچیده‌تر از سایر سیستم‌عامل‌ها است (باید خودتان نصب پایتون را دانلود و بسازید). این به اندازه‌ای که به نظر می‌رسد ترسناک نیست. کفایت دستورات زیر را یکی یکی وارد کنید و منتظر بمانید تا هر کدام کامل شوند (توجه داشته باشید که اگر نسخه‌ای بالاتر از ۳.۱۰ دانلود می‌کنید ممکن است نیاز به تغییر شماره نسخه‌ها داشته باشید):

```

sudo apt update
sudo apt install libffi-dev libssl-dev tk tk-dev
wget https://www.python.org/ftp/python/3.10.0/Python-3.10.0.tar.xz
tar -xvf Python3.10.0.tar.xz
cd Python-3.10.0
./configure --prefix=/usr/local/opt/python-3.10.0
make -j 4
sudo make altinstall

```

گام بعدی، طولانی‌ترین مرحله است. زیرا تمام قوانین مربوط به برنامه پایتون را می‌سازد.

```

python3.10 - Python 3.10.0
File Edit Tabs Help
Compiling '/usr/local/opt/python-3.10.0/lib/python3.10/distutils/command/install_
_scripts.py'...
Compiling '/usr/local/opt/python-3.10.0/lib/python3.10/distutils/command/registe
r.py'...
Compiling '/usr/local/opt/python-3.10.0/lib/python3.10/distutils/command/sdist.p
y'...
Compiling '/usr/local/opt/python-3.10.0/lib/python3.10/distutils/command/upload.
py'...
Compiling '/usr/local/opt/python-3.10.0/lib/python3.10/distutils/config.py'...
Compiling '/usr/local/opt/python-3.10.0/lib/python3.10/distutils/core.py'...
Compiling '/usr/local/opt/python-3.10.0/lib/python3.10/distutils/cygwincompiler
.py'...
Compiling '/usr/local/opt/python-3.10.0/lib/python3.10/distutils/debug.py'...
Compiling '/usr/local/opt/python-3.10.0/lib/python3.10/distutils/dep_util.py'...
Compiling '/usr/local/opt/python-3.10.0/lib/python3.10/distutils/dir_util.py'...
Compiling '/usr/local/opt/python-3.10.0/lib/python3.10/distutils/dist.py'...
Compiling '/usr/local/opt/python-3.10.0/lib/python3.10/distutils/errors.py'...
Compiling '/usr/local/opt/python-3.10.0/lib/python3.10/distutils/extension.py'...
.
Compiling '/usr/local/opt/python-3.10.0/lib/python3.10/distutils/fancy_getopt.py
'...
Compiling '/usr/local/opt/python-3.10.0/lib/python3.10/distutils/file_util.py'...
.
Compiling '/usr/local/opt/python-3.10.0/lib/python3.10/distutils/filelist.py'...

```

شکل ۱-۶: نصب پایتون در Terminal در رزبری پای

با توجه به اینکه کدام نسخه از پایتون را دانلود می‌کنید، خروجی شما ممکن است کمی متفاوت به نظر برسد.

پس از نصب پایتون، باید برنامه‌ای به نام IDLE را به منو اضافه کنید (این کارتان را بعداً آسان‌تر می‌کند):

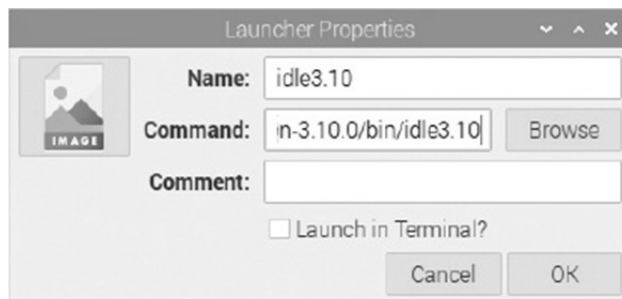
۱. روی آیکن raspberry در سمت چپ بالای صفحه کلیک کنید، سپس روی Preferences and Main Menu Editor کلیک کنید.

۲. در پنجره باز شده، روی Programming کلیک کنید و سپس دکمه New Item را بزنید.

۳. در پنجره Launcher Properties نشان داده شده در شکل ۱-۷، نام idle3.10 را به عنوان دستور وارد کنید و شماره نسخه را در صورت نیاز تغییر دهید:

```
/usr/local/opt/python-3.10.0/bin/idle3.10
```

۴. روی OK کلیک کنید، سپس دوباره OK را در پنجره اصلی ویرایشگر کلیک کنید تا کار تمام شود. سپس می‌توانید به بخش بعدی بروید.



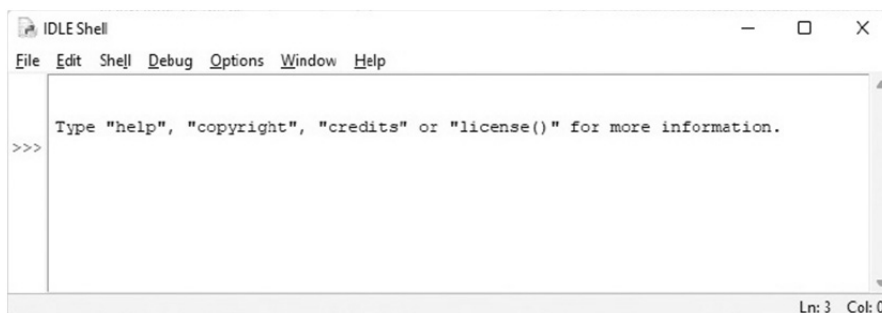
شکل ۷-۱: راه اندازی لانچر در Raspberry Pi

پس از نصب پایتون

با نصب پایتون، بیاید اولین برنامه خود را در IDLE بنویسیم (که به آن Shell نیز می‌گویند). اگر از ویندوز استفاده می‌کنید، idle را در کادر جستجوی ویندوز (پایین سمت چپ صفحه) وارد کنید و IDLE (Python 3.1x 64-bit) را هنگامی که در منوی Best Match دیده شد، انتخاب کنید. اگر از مک استفاده می‌کنید، به Applications ▶ Go بروید و پوشه Python 3.1x را باز کنید تا IDLE را پیدا کنید.

اگر از اوبونتو استفاده می‌کنید، وقتی روی Show Applications کلیک کنید و سپس روی برگه All در پایین کلیک کنید، باید یک ورودی با عنوان IDLE با استفاده از (Python-3.1x) را مشاهده کنید (اگر نتوانستید آنرا ببینید، می‌توانید IDLE را در کادر جستجو نیز وارد کنید).

اگر از رزبری پای استفاده می‌کنید، روی آیکن raspberry در سمت چپ بالای صفحه کلیک کنید، روی Programming کلیک کنید و سپس idle3.1x را از لیست نمایش داده شده انتخاب کنید. وقتی IDLE را باز می‌کنید، باید پنجره‌ای مانند این را ببینید:



شکل ۸-۱: IDLE Shell در ویندوز

این همان پوسته پایتون است، که بخشی از محیط توسعه یکپارچه پایتون می‌باشد. به سه علامت بزرگتر از (<<<) پرامپت (prompt) می‌گویند.

بیا ببینیم چند دستور را در پرامپت وارد کنیم که با دستور زیر شروع می‌شود:

```
>>> print("Hello World")
```

مطمئن شوید که کوتیشن‌های دوگانه (" ") را درج می‌کنید. وقتی تایپ خط تمام شد، ENTER را روی صفحه کلید خود فشار دهید.

اگر دستور را به درستی وارد کرده باشید، باید چیزی شبیه به این را ببینید:

```
>>> print("Hello World")
Hello World
>>>
```



پرامپت باید دوباره ظاهر شود تا به شما اطلاع دهد که پوسته پایتون آماده پذیرش دستورات بیشتر است.

تبریک می‌گوییم! شما اولین برنامه پایتون خود را ایجاد کرده‌اید. `print` یک نوع دستور پایتون است که تابع نامیده می‌شود و هر آنچه را که داخل پرانتز است روی صفحه نمایش می‌دهد. در اصل، شما دستورالعملی به رایانه داده‌اید که کلمات "Hello World" را نمایش دهد (دستورالعملی که هم شما و هم رایانه می‌توانید آن را بفهمید).

ذخیره کردن برنامه‌های پایتون

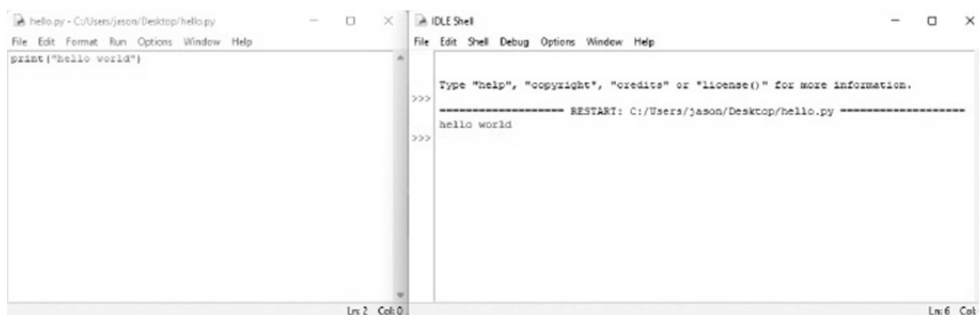
اگر نیاز باشد هر بار که می‌خواهید از برنامه‌های پایتون استفاده کنید آنها را بازنویسی کنید، فایده زیادی نخواهند داشت. مطمئناً، بازنویسی برنامه‌های کوتاه ممکن است خوب باشد، اما یک برنامه بزرگ، مانند یک واژه پرداز، می‌تواند حاوی میلیون‌ها خط کد باشد. اگر تمام آن را چاپ کنید، می‌توانید بیش از ۱۰۰۰۰۰ صفحه داشته باشید. فقط تصور کنید بخواهید آن دسته عظیم کاغذ را به خانه ببرید. بهتر است امیدوار باشید که با وزش باد شدید روبرو نشوید!

خوشبختانه، ما می‌توانیم برنامه‌های خود را برای استفاده در آینده ذخیره کنیم. برای ایجاد و ذخیره یک برنامه جدید، IDLE را باز کرده و `File ▶ New Window` را انتخاب کنید. یک پنجره خالی با `Untitled` در نوار منو ظاهر می‌شود. کد زیر را در پوسته جدید وارد کنید:

```
>>> print("Hello World")
```

اکنون **File ▶ Save** را انتخاب کنید. وقتی نام فایل از شما خواسته شد، `hello.py` را وارد کنید و فایل را در دستکاپ خود ذخیره کنید. سپس **Run ▶ Run Module** را انتخاب کنید.

با هر شانسی، برنامه ذخیره شده شما باید مانند زیر اجرا شود:



شکل ۹-۱: Hello World در IDLE

اکنون، اگر پنجره پوسته را ببندید اما پنجره `hello.py` را باز بگذارید و سپس **Run ▶ Run Module** را انتخاب کنید، پوسته پایتون باید دوباره ظاهر شود و برنامه شما باید دوباره اجرا شود (برای باز کردن مجدد پوسته پایتون بدون اجرای برنامه، **Run ▶ Python Shell** را انتخاب کنید).

پس از اجرای کد، آیکن جدیدی در دستکاپ خود با عنوان `hello.py` پیدا خواهید کرد. اگر روی آیکن دوبار کلیک کنید، یک پنجره سیاه به طور خلاصه ظاهر می‌شود و سپس ناپدید می‌شود. چی شد؟

می‌بینید که کنسول خط فرمان پایتون (شبهه به پوسته) شروع به کار می‌کند، `Hello World` را چاپ می‌کند و سپس خارج می‌شود. اگر دید سرعتی ابرقهرمانی داشتید و توانستید پنجره را پیش از بسته شدن ببینید، این چیزی است که ظاهر می‌شود:



شکل ۱۰-۱: Hello World در کنسول

توجه: به سیستم‌عامل شما بستگی دارد، این ممکن است کار نکند (یا ممکن است با استفاده از نسخه‌ای متفاوت از نسخه پایتونی که ما نصب کرده‌ایم اجرا شود).

علاوه بر منوها، می‌توانید از میانبرهای صفحه کلید برای ایجاد یک پنجره پوسته جدید، ذخیره یک فایل و اجرای یک برنامه استفاده کنید:

در ویندوز، اوبونتو و رزبری‌پای، CTRL-N را برای ایجاد یک پنجره پوسته جدید، CTRL-S را برای ذخیره فایل پس از اتمام ویرایش و F5 را برای اجرای برنامه فشار دهید.

در macOS، COMMAND-N را فشار دهید تا یک پنجره پوسته جدید ایجاد کنید، COMMAND-S را برای ذخیره فایل خود فشار دهید، و کلید تابع (Fn) کیبورد را نگه دارید و برای اجرای برنامه، F5 را فشار دهید.

آنچه در این فصل یاد گرفتید

در این فصل با برنامه Hello World شروع کردیم. برنامه‌ای که تقریباً هر کسی که برنامه‌نویسی را یاد می‌گیرد با آن شروع می‌کند. در فصل بعد، کارهای مفیدتری با پوسته پایتون انجام می‌دهیم.

فصل ۲

محاسبات و متغیرها



اکنون که پایتون را نصب کرده‌اید و می‌دانید چگونه پوسته پایتون را راه‌اندازی کنید، آماده کار کردن با آن هستید. با محاسبات ساده شروع می‌کنیم و سپس نحوه استفاده از متغیرها را یاد می‌گیریم. متغیرها راهی برای ذخیره کردن چیزها در یک برنامه کامپیوتری هستند و می‌توانند به شما در نوشتن برنامه‌های مفید کمک کنند.

محاسبه با پایتون

معمولاً وقتی از شما خواسته می‌شود حاصل ضرب دو عدد مانند 8×3.57 را پیدا کنید، از ماشین حساب یا مداد و کاغذ استفاده می‌کنید. خوب، چطور باید از پوسته پایتون برای انجام محاسبه استفاده کنید؟ بیایید آن را امتحان کنیم.

با دوبار کلیک کردن روی آیکن IDLE روی دسکتاپ خود، پوسته پایتون را باز کنید یا اگر از اوبونتو استفاده می‌کنید، روی آیکن IDLE در منوی برنامه‌ها کلیک کنید. در پرامپت، این محاسبه را وارد کنید:

```
>>> 8 * 3.57
28.56
```

هنگام وارد کردن محاسبه ضرب در پایتون، باید به جای علامت ضرب (\times) از علامت ستاره (*) استفاده کنید.

چطور است یک معادله را امتحان کنیم که کمی مفیدتر باشد؟

فرض کنید در حیاط خانه خود حفاری می‌کنید و یک کیسه با ۲۰ سکه طلا کشف می‌کنید. روز بعد، مخفیانه به زیرزمین می‌روید و سکه‌ها را داخل دستگاه تکثیر کننده با نیروی بخار که پدر بزرگتان اختراع کرده است می‌گذارید (خوشبختانه، فقط می‌توانید ۲۰ سکه را داخل آن قرار دهید). صدای سوت و فیس فیس می‌شنوید و چند ساعت بعد، ۱۰ سکه درخشان دیگر به بیرون پرت می‌شود!

اگر به مدت یک سال هر روز این کار را انجام دهید، چند سکه در صندوق گنج خواهید داشت؟ روی کاغذ، معادلات ممکن است به این صورت باشد:

$$10 \times 365 = 3650$$

$$3650 + 20 = 3670$$

مطمئناً انجام این محاسبات با استفاده از ماشین حساب یا روی کاغذ به اندازه کافی آسان است، اما می‌توانیم همه این محاسبات را با پوسته پایتون نیز انجام دهیم. ابتدا ۱۰ سکه را در ۳۶۵ روز در سال ضرب می‌کنیم تا عدد ۳۶۵۰ به دست آید. سپس ۲۰ سکه اصلی را اضافه می‌کنیم تا عدد ۳۶۷۰ به دست آید.

```
>>> 10 * 365
3650
>>> 3650 + 20
3670
```

اکنون اگر یک کلاغ طلای براقی را که در اتاق خواب شما است ببیند و هر هفته پرواز کند و موفق شود سه سکه را بدزدد، چه؟ در پایان سال چند سکه باقی می‌ماند؟ در اینجا نحوه محاسبه در پوسته پایتون آمده است:

```
>>> 3 * 52
156
>>> 3670 - 156
3514
```

ابتدا ۳ سکه را در ۵۲ هفته در سال ضرب می‌کنیم. نتیجه ۱۵۶ است. این عدد را از تعداد کل سکه‌های خود (۳۶۷۰) کم می‌کنیم، که به ما می‌گوید در پایان سال ۳۵۱۴ سکه باقی می‌ماند.

اگرچه می‌توانید این محاسبه را به راحتی با ماشین حساب انجام دهید، اما کار با آن در پوسته پایتون برای یادگیری نوشتن برنامه‌های کامپیوتری ساده مفید است. در این کتاب، یاد خواهید گرفت که چگونه این ایده‌ها را برای نوشتن برنامه‌هایی که حتی مفیدتر هستند، گسترش دهید.