

THIRD EDITION

PYTHON CRASH COURSE

**A HANDS-ON, PROJECT-BASED
INTRODUCTION TO PROGRAMMING**

ERIC MATTHES

**OVER
1,500,000
COPIES SOLD**



ELOGIOS PARA CURSO CRASH DE PYTHON

“Ha sido interesante ver a No Starch Press producir futuros clásicos que deberían estar junto a los libros de programación más tradicionales. Python Crash Course es uno de esos libros”.

—Greg Laden, Blogs de ciencia

"Aborda algunos proyectos bastante complejos y los presenta de una manera coherente, lógica y agradable que atrae al lector hacia el tema".

—Revista Círculo Completo

“Bien presentado con buenas explicaciones de los fragmentos de código. El libro trabaja contigo, un pequeño paso a la vez, creando un código más complejo y explicando lo que está sucediendo en todo momento”.

—Reseñas de hojear

“¡Aprender Python con el curso intensivo de Python fue una experiencia extremadamente positiva! Una gran elección si eres nuevo en Python”.

—Mikke se dedica a codificar

“Hace lo que dice y lo hace realmente bien. . . . Presenta una gran cantidad de ejercicios útiles, así como tres proyectos desafiantes y entretenidos”.

—RealPython.com

"Una introducción rápida pero completa a la programación con Python, Python Crash Course es otro excelente libro para agregar a tu biblioteca y ayudarte a dominar finalmente Python".

—TutorialEdge.net

“Una opción brillante para principiantes sin experiencia en codificación. Si está buscando una introducción sólida y sencilla a este lenguaje tan profundo, debo recomendar este libro”.

—WhatPixel.com

"Contiene literalmente todo lo que necesitas saber sobre Python y aún más".

—FireBearStudio.com

"Si bien el curso intensivo de Python utiliza Python para enseñarle a codificar, también enseña habilidades de programación limpias que se aplican a la mayoría de los demás lenguajes".

—Geek de los Grandes Lagos

PYTHONCRASH CURSO

3ª EDICIÓN

AH ys - En , Proyecto basado
Introducción a la programación

por Eric Matthes



**no starch
press**

San Francisco

CURSO CRASH DE PYTHON, 3ª EDICIÓN. Copyright © 2023 por Eric Matthes.

Reservados todos los derechos. Ninguna parte de este trabajo puede reproducirse o transmitirse de ninguna forma ni por ningún medio, electrónico o mecánico, incluidas fotocopias, grabaciones o cualquier sistema de almacenamiento o recuperación de información, sin el permiso previo por escrito del propietario de los derechos de autor y del editor.

Primera impresión

26 25 24 23 22 1 2 3 4 5

ISBN-13: 978-1-7185-0270-3 (impreso)
ISBN-13: 978-1-7185-0271-0 (libro electrónico)

Editorial: William Pollock
Editor jefe: Jill Franklin
Editor de producción: Jennifer Kepler
Editora de desarrollo: Eva Morrow
Ilustrador de portada: Josh Ellingson
Diseño de interiores: Estudios Octopod
Revisor técnico: Kenneth Love

Editor: Doug McNair
Compositor: Jeff Lytle, Happenstance Type-O-Rama
Corrector: Scout Festa

Para obtener información sobre distribución, ventas al por mayor, ventas corporativas o traducciones, comuníquese con No Starch Press, Inc. directamente en info@nostarch.com o:

Sin almidón Press, Inc.
245 Calle 8, San Francisco, CA 94103
teléfono: 1.415.863.9900
www.nostarch.com

La Biblioteca del Congreso ha catalogado la primera edición de la siguiente manera:

Matthes, Eric, 1972-
Curso intensivo de Python: una introducción práctica a la programación basada en proyectos / por Eric Matthes.
páginas cm
Incluye índice.

Resumen: "Una introducción a la programación en Python basada en proyectos, con ejercicios. Cubre conceptos generales de programación, fundamentos de Python y resolución de problemas. Incluye tres proyectos: cómo crear un videojuego simple, usar técnicas de visualización de datos para crear gráficos y tablas, y crear una aplicación web interactiva"--

Proporcionado por el editor
ISBN 978-1-59327-603-4 - ISBN 1-59327-603-6
1. Python (Lenguaje de programación informática) I. Título.
QA76.73.P98M38 2015
005.13'3--dc23

2015018135

No Starch Press y el logotipo de No Starch Press son marcas comerciales registradas de No Starch Press, Inc. Otros nombres de productos y empresas mencionados aquí pueden ser marcas comerciales de sus respectivos propietarios. En lugar de utilizar un símbolo de marca registrada cada vez que aparece un nombre de marca registrada, utilizamos los nombres solo de manera editorial y para beneficio del propietario de la marca, sin intención de infringir la marca.

La información contenida en este libro se distribuye "tal cual", sin garantía. Si bien se han tomado todas las precauciones en la preparación de este trabajo, ni el autor ni No Starch Press, Inc. tendrán responsabilidad alguna ante ninguna persona o entidad con respecto a cualquier pérdida o daño causado o presuntamente causado directa o indirectamente por el información contenida en el mismo.

Para mi padre, que siempre se tomaba el
tiempo para responder mis preguntas sobre
programación, y para Ever, que recién comienza
a hacerme sus preguntas.

Sobre el Autor

Eric Matthes fue profesor de matemáticas y ciencias en una escuela secundaria durante 25 años, e impartía clases de introducción a Python siempre que encontraba una manera de incluirlas en el plan de estudios. Eric ahora es escritor y programador a tiempo completo y participa en varios proyectos de código abierto. Sus proyectos tienen una amplia gama de objetivos, desde ayudar a predecir la actividad de deslizamientos de tierra en regiones montañosas hasta simplificar el proceso de implementación de proyectos Django. Cuando no está escribiendo o programando, le gusta escalar montañas y pasar tiempo con su familia.

Acerca del revisor técnico

Kenneth Love vive en el noroeste del Pacífico con su familia y sus gatos. Kenneth es un veterano programador de Python, colaborador de código abierto, profesor y conferencista.

CONTENIDOS BREVES

Prefacio a la Tercera Edición.	xxvii
Agradecimientos.	xxxi
Introducción	xxxiii
 PARTE I: FUNDAMENTOS	 1
Capítulo 1: Primeros pasos.	3
Capítulo 2: Variables y tipos de datos simples.	15
Capítulo 3: Presentación de listas.	33
Capítulo 4: Trabajar con listas.	49
Capítulo 5: Declaraciones if.	71
Capítulo 6: Diccionarios.	91
Capítulo 7: Entrada del usuario y bucles while.	113
Capítulo 8: Funciones.	129
Capítulo 9: Clases.	157
 Capítulo 10: Archivos y excepciones.	 183
Capítulo 11: Probando su código.	209
 PARTE II: PROYECTOS	 225
Capítulo 12: Un barco que dispara balas.	227
Capítulo 13: ¡Extraterrestres!	255
Capítulo 14: Puntuación.	277
Capítulo 15: Generación de datos.	301
 Capítulo 16: Descarga de datos.....	 329
Capítulo 17: Trabajar con API.	355

Capítulo 18: Primeros pasos con Django. 373

Capítulo 19: Cuentas de usuario. 403

Capítulo 20: Diseño e implementación de una aplicación. 433

Apéndice A: Instalación y solución de problemas. 463

Apéndice B: Editores de texto e IDE. 469

Apéndice C: Obtener ayuda. 477

Apéndice D: Uso de Git para el control de versiones. 483

Apéndice E: Solución de problemas de implementaciones. 493

Índice. 503

CONTENIDOS EN DETALLE

PREFACIO A LA TERCERA EDICIÓN	xxvii
EXPRESIONES DE GRATITUD	xxxix
INTRODUCCIÓN	xxxiii
¿Para quién es este libro?	xxxiii
¿Qué puedes esperar aprender?	xxxiii
Recursos en línea	xxxv
¿Por qué Python?	xxxvi
PARTE I: FUNDAMENTOS	1
1	
EMPEZANDO	3
Configurando su entorno de programación.	3
Versiones de Python.	4
Ejecución de fragmentos de código	4
Python Acerca del editor de	4
código VS Python en diferentes sistemas operativos.	5
Pitón en Windows.	5
Python en macOS.	7
Pitón en Linux.	8
Ejecución de un programa Hola Mundo.	9
Instalación de la extensión Python para VS	9
Code ejecutando hello_world.py.	10
Solución de problemas	10
Ejecutar programas Python desde una terminal.	11
En Windows.	12
En macOS y Linux	12
Ejercicio 1-1: python.org.	13
Ejercicio 1-2: Errores tipográficos de Hola mundo.	13
Ejercicio 1-3: Habilidades Infinitas.	13
Resumen	13
2	
VARIABLES Y TIPOS DE DATOS SIMPLES	15
Qué sucede realmente cuando ejecutas hello_world.py.	15
Variables.	dieciséis
Nombrar y usar variables.	17
Evitar errores de nombre al utilizar variables.	17
Las variables son etiquetas.	18
Ejercicio 2-1: Mensaje sencillo.	19
Ejercicio 2-2: Mensajes sencillos.	19

Cuerdas. . . : 19

 Cambiar mayúsculas y minúsculas en una cadena con métodos. . . : 20

 Uso de variables en cadenas. . . : 20

 Agregar espacios en blanco a cadenas con tabulaciones o nuevas líneas. 21

 Eliminación de espacios en blanco. 22

 Eliminación de prefijos. . . : 23

 Evitar errores de sintaxis con cadenas. 24

 Ejercicio 2-3: Mensaje personal. 25

 Ejercicio 2-4: Nombre de casos. 25

 Ejercicio 2-5: Cita célebre. 25

 Ejercicio 2-6: Cita célebre 2. 25

 Ejercicio 2-7: Eliminación de nombres. 25

 Ejercicio 2-8: Extensiones de archivos. 25

Números. . . : 26

 Enteros. . . . : 26

 Flotadores. . . : 26

 Enteros y flotantes. . . : 27

 Guiones bajos en números. . . : 28

 Asignación múltiple. . . . : 28

 Constantes. . . : 28

 Ejercicio 2-9: Número ocho. 29

 Ejercicio 2-10: Número favorito. 29

Comentarios. . . : 29

 ¿Cómo se escriben comentarios? . . . : 29

 ¿Qué tipo de comentarios debería escribir? . . . : 29

 Ejercicio 2-11: Agregar comentarios. 30

El Zen de Python. . . : 30

 Ejercicio 2-12: Zen de Python. 31

Resumen . . . : 32

3

PRESENTANDO LISTAS

33

¿Qué es una lista? : 33

 Accediendo a elementos de una lista: . . : 34

 Las posiciones del índice comienzan en 0, no en 1. : 34

 Usar valores individuales de una lista. . . : 35

 Ejercicio 3-1: Nombres. 36

 Ejercicio 3-2: Saludos. 36

 Ejercicio 3-3: Su propia lista. 36

Modificar, agregar y eliminar elementos. . . : 36

 Modificar elementos en una lista. . . : 36

 Agregar elementos a una lista. . . : 37

 Eliminar elementos de una lista. . . : 38

 Ejercicio 3-4: Lista de invitados. 41

 Ejercicio 3-5: cambiar la lista de invitados. 42

 Ejercicio 3-6: Más invitados. 42

 Ejercicio 3-7: Lista de invitados reducida. 42

Organizar una lista. . : 42

 Ordenar una lista de forma permanente con el método sort(). . . : 43

 Ordenar una lista temporalmente con la función sorted(). . . : 43

 Imprimir una lista en orden inverso. 44

Encontrar la longitud de una lista. . 44

Ejercicio 3-8: Ver el mundo. 45

Ejercicio 3-9: Invitados a cenar. 45

Ejercicio 3-10: cada función. 45

Evitar errores de índice al trabajar con listas. 46

Ejercicio 3-11: Error intencional. 47

Resumen 47

4

TRABAJAR CON LISTAS 49

Recorrer una lista completa. 49

Una mirada más cercana al bucle. 50

Hacer más trabajo dentro de un bucle for. 51

Hacer algo después de un bucle for. 52

Evitar errores de sangría. . . 53

Olvidarse de sangrar. . . 53

Olvidar aplicar sangría a líneas adicionales. . . . 54

Sangrar innecesariamente. . . . 54

Sangrar innecesariamente después del bucle. . . . 55

Olvidando el Colón. . . . 55

Ejercicio 4-1: Pizzas. 56

Ejercicio 4-2: Animales. 56

Hacer listas numéricas. 56

Usando la función range(). . . . 57

Usando range() para hacer una lista de números. . . . 58

Estadísticas simples con una lista de números. . . . 59

Lista de comprensiones. . . . 59

Ejercicio 4-3: Contar hasta veinte. 60

Ejercicio 4-4: Un millón. 60

Ejercicio 4-5: Sumar un millón. 60

Ejercicio 4-6: Números impares. 60

Ejercicio 4-7: Tres. 60

Ejercicio 4-8: Cubos. 60

Ejercicio 4-9: Comprensión de cubos. 60

Trabajar con parte de una lista. 61

Cortar una lista. . . 61

Recorriendo un corte. . . 62

Copiar una lista. . . 63

Ejercicio 4-10: Rebanadas. sesenta y cinco

Ejercicio 4-11: Mis pizzas, tus pizzas. sesenta y cinco

Ejercicio 4-12: Más bucles. sesenta y cinco

Tuplas. sesenta y cinco

Definiendo una tupla. . . . sesenta y cinco

Recorriendo todos los valores de una tupla. . . . 66

Escribir sobre una tupla. . . . 67

Ejercicio 4-13: Buffet. 67

Diseñar su código. . . 68

La guía de estilo. . . . 68

Sangría. . . 68

Longitud de la línea 69

Líneas en blanco 69

Otras pautas de estilo. . . 69	
Ejercicio 4-14: PEP 8	70
Ejercicio 4-15: Revisión de código.	70
Resumen	70
 5	
DECLARACIONES SI	71
Un ejemplo sencillo. . . 72	
Pruebas condicionales. 72	
Comprobando la igualdad. 72	
Ignorar el caso al verificar la igualdad. 73	
Comprobando la desigualdad. 74	
Comparaciones numéricas. 74	
Comprobación de múltiples condiciones. 75	
Comprobar si un valor está en una lista. 76	
Comprobar si un valor no está en una lista. 76	
Expresiones booleanas. 77	
Ejercicio 5-1: Pruebas condicionales. 77	
Ejercicio 5-2: Más pruebas condicionales. 78	
Si declaraciones. 78	
Declaraciones if simples. 78	
Declaraciones if-else. 79	
La cadena if-elif-else. 80	
Usando múltiples bloques elif. 81	
Omitiendo el bloque else. 82	
Prueba de múltiples condiciones. 82	
Ejercicio 5-3: Colores alienígenas n.º 1. 84	
Ejercicio 5-4: Colores alienígenas n.º 2. 84	
Ejercicio 5-5: Colores alienígenas n.º 3. 84	
Ejercicio 5-6: Etapas de la vida. 84	
Ejercicio 5-7: Fruta favorita. 85	
Uso de sentencias if con listas. 85	
Comprobación de artículos especiales. 85	
Comprobar que una lista no esté vacía. 86	
Usando múltiples listas. 87	
Ejercicio 5-8: Hola administrador. 88	
Ejercicio 5-9: Sin usuarios. 88	
Ejercicio 5-10: Comprobación de nombres de usuario. 88	
Ejercicio 5-11: Números ordinales. 88	
Diseñar sus declaraciones if. 89	
Ejercicio 5-12: Aplicar estilo a declaraciones if. 89	
Ejercicio 5-13: Tus ideas. 89	
Resumen	89
 6	
DICCIONARIOS	91
Un diccionario sencillo.	92
Trabajar con diccionarios.	92
Acceso a valores en un diccionario.	92
Agregar nuevos pares clave-valor.	93
Comenzando con un diccionario vacío.	94

Modificar valores en un diccionario. . . 94	
Eliminación de pares clave-valor. . . 96	
Un diccionario de objetos similares. . . 96	
Usando get() para acceder a valores: . . 97	
Ejercicio 6-1: Persona.	98
Ejercicio 6-2: Números favoritos.	98
Ejercicio 6-3: Glosario.	99
Recorriendo un diccionario. . . 99	
Recorriendo todos los pares clave-valor. . . 99	
Recorriendo todas las claves de un diccionario.....	101
Recorrer las claves de un diccionario en un orden particular. . . 102	
Recorriendo todos los valores en un diccionario. . 103	
Ejercicio 6-4: Glosario 2	104
Ejercicio 6-5: Ríos.	105
Ejercicio 6-6: sondeo.	105
Anidación. . . 105	
Una lista de diccionarios. . . 105	
Una lista en un diccionario. . . 108	
Un diccionario en un diccionario. : 110	
Ejercicio 6-7: Personas.	111
Ejercicio 6-8: Mascotas.	111
Ejercicio 6-9: Lugares favoritos.	111
Ejercicio 6-10: Números favoritos.	111
Ejercicio 6-11: Ciudades.	111
Ejercicio 6-12: Extensiones.	111
Resumen	111
 7	
ENTRADA DEL USUARIO Y BUCLES WHILE	113
Cómo funciona la función input(). 114	
Escribir indicaciones claras. 114	
Usando int() para aceptar entradas numéricas. . . 115	
El operador de módulo. . 116	
Ejercicio 7-1: Coche de alquiler.	117
Ejercicio 7-2: Asientos en un restaurante.	117
Ejercicio 7-3: Múltiplos de diez.	117
Presentamos los bucles while.	117
El bucle while en acción. . 117	
Permitir que el usuario elija cuándo salir. . 118	
Usando una bandera. : 120	
Usando break para salir de un bucle. . 121	
Usando continuar en un bucle.	122
Evitando bucles infinitos. 122	
Ejercicio 7-4: Ingredientes para pizza.	123
Ejercicio 7-5: Entradas al cine.	123
Ejercicio 7-6: Tres salidas.	123
Ejercicio 7-7: Infinito.	123
Usando un bucle while con listas y diccionarios. . 124	
Mover elementos de una lista a otra. 124	
Eliminación de todas las instancias de valores específicos de una lista. . . 125	
Llenar un diccionario con entradas del usuario: . 125	
Ejercicio 7-8: Deli	127

Ejercicio 7-9: Sin pastrami. 127

Ejercicio 7-10: Vacaciones de ensueño. 127

Resumen 127

8

FUNCIONES129

Definición de una función. . . 130

Pasar información a una función. . 130

Argumentos y parámetros. . . 131.

Ejercicio 8-1: Mensaje. 131

Ejercicio 8-2: Libro favorito. 131

Pasar argumentos. . 131

Argumentos posicionales. . . 132

Argumentos de palabras clave. . : 133

Valores predeterminados 134

Llamadas a funciones equivalentes..... 135

Evitar errores de argumentación. . . 136

Ejercicio 8-3: Camiseta. 136

Ejercicio 8-4: Camisas grandes. 137

Ejercicio 8-5: Ciudades. 137

Valores de retorno. . 137

Devolver un valor simple. . 137

Hacer que un argumento sea opcional. . 138

Devolver un diccionario. . 139

Usando una función con un bucle while. . 140

Ejercicio 8-6: Nombres de ciudades. 141

Ejercicio 8-7: Álbum. 142

Ejercicio 8-8: Álbumes de usuario. 142

Pasando una lista. 142

Modificar una lista en una función. . : 143

Evitar que una función modifique una lista. . 145

Ejercicio 8-9: Mensajes. 146

Ejercicio 8-10: Envío de mensajes. 146

Ejercicio 8-11: Mensajes archivados. 146

Pasar un número arbitrario de argumentos. 146

Mezclando argumentos posicionales y arbitrarios. . 147.

Uso de argumentos de palabras clave arbitrarias..... 148

Ejercicio 8-12: Sándwiches. 149

Ejercicio 8-13: Perfil de usuario. 149

Ejercicio 8-14: Coches. 149

Almacenamiento de funciones en módulos. . 149

Importación de un módulo completo: . 150

Importación de funciones específicas: . 151

Usar as para darle un alias a una función. . . 151

Usar as para darle un alias a un módulo. . . 152

Importación de todas las funciones de un módulo. . : 152

Funciones de estilo. . . 153

Ejercicio 8-15: Modelos de impresión. 154

Ejercicio 8-16: Importaciones. 154

Ejercicio 8-17: Funciones de estilo. 154

Resumen . . 154

9

CLASES

157

Creación y uso de una clase.	158
Creando la clase de perro.	158
El método __init__().	159
Crear una instancia a partir de una clase.	159
Ejercicio 9-1: Restaurante.	162
Ejercicio 9-2: Tres restaurantes.	162
Ejercicio 9-3: Usuarios.	162
Trabajar con clases e instancias.	162
La clase de coche.	162
Establecer un valor predeterminado para un atributo.	163
Modificación de valores de atributos.	164
Ejercicio 9-4: Número atendido.	166
Ejercicio 9-5: intentos de inicio de sesión.	167
Herencia	167
El método __init__() para una clase secundaria.	167
Definición de atributos y métodos para la clase secundaria.	169
Anulación de métodos de la clase principal.	170
Instancias como atributos.	170
Modelado de objetos del mundo real.	172
Ejercicio 9-6: Puesto de helados.	173
Ejercicio 9-7: Administrador.	173
Ejercicio 9-8: Privilegios.	173
Ejercicio 9-9: Actualización de la batería.	173
Importación de clases.	173
Importando una sola clase.	174
Almacenamiento de varias clases en un módulo.	175
Importación de varias clases desde un módulo.	176
Importación de un módulo completo.	176
Importar todas las clases desde un módulo.	177
Importar un módulo a un módulo.	177
Usando alias.	178
Encontrar su propio flujo de trabajo.	179
Ejercicio 9-10: Restaurante importado.	179
Ejercicio 9-11: Administrador importado.	179
Ejercicio 9-12: Múltiples módulos.	179
La biblioteca estándar de Python.	179
Ejercicio 9-13: Datos.	180
Ejercicio 9-14: Lotería.	180
Ejercicio 9-15: Análisis de lotería.	180
Ejercicio 9-16: Módulo Python de la semana.	180
Clases de estilismo.	181
Resumen	181

10

ARCHIVOS Y EXCEPCIONES

183

Lectura de un archivo.	184
Leer el contenido de un archivo.	184
Rutas de archivo relativas y absolutas.	186
Accediendo a las líneas de un archivo.	186

Trabajar con el contenido de un archivo: : 187 · · · · ·

Archivos grandes: un millón de dígitos.: 188 · · · · ·

¿Tu cumpleaños está contenido en Pi? : 189 · · · · ·

Ejercicio 10-1: Aprender Python. 189

Ejercicio 10-2: Aprendizaje C 190

Ejercicio 10-3: Código más simple. 190

Escribir en un archivo: : 190 · · · · ·

Escribir una sola línea. . . 190 · · · · ·

.....

Escribir varias líneas. 191 192

Ejercicio 10-4: Libro de visitas. 192

Excepciones. : 192 · · · · ·

Manejo de la excepción ZeroDivisionError. . . 192 · · · · ·

Usando bloques try-except. . . 193 · · · · ·

Uso de excepciones para evitar fallas. . . 193 · · · · ·

El otro bloque. . . 194 · · · · ·

Manejo de la excepción FileNotFoundError. . . 195 · · · · ·

Analizando texto. . 196 · · · · ·

Trabajar con varios archivos. . 197 · · · · ·

Fallando en silencio.: 198 · · · · ·

Decidir qué errores informar. . . 199 · · · · ·

Ejercicio 10-6: Suma. 200

Ejercicio 10-7: Calculadora de sumas. 200

Ejercicio 10-8: Perros y gatos. 200

Ejercicio 10-9: Perros y gatos silenciosos. 200

Ejercicio 10-10: Palabras comunes. 200

Almacenamiento de datos : 201 · · · · ·

Usando json.dumps() y json.loads(). . . 201 · · · · ·

Guardar y leer datos generados por el usuario. . . 202 · · · · ·

Refactorización. : 204 · · · · ·

Ejercicio 10-11: Número favorito. 206

Ejercicio 10-12: Número favorito recordado. 206

Ejercicio 10-13: Diccionario del usuario. 206

Ejercicio 10-14: verificar usuario. 206

Resumen . . 207 · · · · ·

11

PROBANDO SU CÓDIGO 209

Instalando pytest con pip. . 210 · · · · ·

Actualizando pip. : 210 · · · · ·

Instalando pytest. 211 · · · · ·

Probar una función. . 211 · · · · ·

Pruebas unitarias y casos de prueba. 212.

Una prueba de aprobación. : 212 · · · · ·

Ejecutando una prueba. : 213 · · · · ·

Una prueba fallida. . 214 · · · · ·

Responder a una prueba fallida. : 215 · · · · ·

Agregar nuevas pruebas: : 216 · · · · ·

Ejercicio 11-1: Ciudad, País. 217

Ejercicio 11-2: Población. 217

Probar una clase. . :217

 Una variedad de afirmaciones: : 217

 Una clase para probar. . 218

 Probando la clase AnonymousSurvey. . 220

 Usando accesorios. . 221

 Ejercicio 11-3: Empleado. 223

Resumen . . 223

PARTE II: PROYECTOS 225

12

UN BARCO QUE DISPARA BALAS 227

Planificando su proyecto. 228

Instalando Pygame. 228

Iniciando el proyecto del juego. 229

 Crear una ventana de Pygame y responder a la entrada del usuario. 229

 Controlar la velocidad de fotogramas. 230

 Configuración del color de fondo. 231

 Creando una clase de configuración. 232

Agregar la imagen del barco. 233

 Creando la clase de barco. 234

 Llevando el barco a la pantalla. 235

Refactorización: los métodos _check_events() y _update_screen()..... 237

 El método _check_events(). . 237

 El método _update_screen(). . 237

 Ejercicio 12-1: Cielo azul. 238

 Ejercicio 12-2: Personaje del juego. 238

Pilotando el barco. .238

 Responder a una pulsación de tecla. : : 238

 Permitiendo el movimiento continuo. . . 239

 Moviéndose tanto hacia la izquierda como hacia la derecha. : 241

 Ajuste de la velocidad del barco. . 242

 Limitar el alcance del barco. . 243

 Refactorizando _check_events() . 244

 Presionando Q para salir. 244

 Ejecutando el juego en modo de pantalla completa: : 245

Un resumen rápido.: :245

 alien_invasion.py. . 246

 configuración.py.: 246

 barco.py. . . 246

 Ejercicio 12-3: Documentación de Pygame. 246

 Ejercicio 12-4: Cohete. 246

 Ejercicio 12-5: Claves. 246

Disparando balas. . 247

 Agregar la configuración de viñetas: : 247

 Creando la clase Bullet. . 247

 Almacenamiento de viñetas en un grupo. . . 248

 Disparando balas: : 249

 Eliminación de viñetas antiguas. : 250

Limitar el número de balas.	251
Creando el método _update_bullets().	252
Ejercicio 12-6: Tirador lateral.	253
Resumen	253
 13	
¡EXTRANJEROS!	255
Revisando el Proyecto.	256
Creando el primer alienígena.	256
Creando la clase alienígena.	257
Creando una instancia del alienígena.	257
Construyendo la flota alienígena.	259
Creando una fila de alienígenas.	259
Refactorizando _create_fleet().	260
Agregar filas.	261
Ejercicio 13-1: Estrellas.	263
Ejercicio 13-2: Mejores estrellas.	263
Hacer que la flota se mueva.	263
Moviendo a los extraterrestres hacia la derecha.	263
Creación de configuraciones para la dirección de la flota.	264
Comprobar si un extraterrestre ha llegado al borde.	265
"Dejar caer la flota y cambiar de dirección".	265
Ejercicio 13-3: Gotas de lluvia.	266
Ejercicio 13-4: Lluvia constante.	266
Disparar a extraterrestres.	266
Detección de colisiones de balas.	267
"Hacer balas más grandes para realizar pruebas".	268
Repoblación de la Flota.	268
Acelerando las balas.	269
Refactorizando _update_bullets().	269
Ejercicio 13-5: Tirador lateral, parte 2.	270
Terminando el juego.	270
Detección de colisiones de naves alienígenas.	270
Respondiendo a las colisiones de naves alienígenas.	271
Extraterrestres que llegan al final de la pantalla.	273
¡Juego terminado!	274
Identificar cuándo deben ejecutarse partes del juego.	275
Ejercicio 13-6: Fin del juego.	275
Resumen	275
 14	
PUNTUACIÓN	277
Agregar el botón Reproducir.	278
Creando una clase de botón.	278
Dibujando el botón a la pantalla.	279
Comenzando el juego.	281
Restablecer el juego.	281
Desactivar el botón Reproducir.	282
Ocultar el cursor del mouse.	282
Ejercicio 14-1: Presione P para reproducir.	283
Ejercicio 14-2: Práctica de tiro.	283

Subir de nivel . . . 283

 Modificación de la configuración de velocidad. . . 283

 Restablecer la velocidad. . . 285

 Ejercicio 14-3: Práctica de tiro desafiante. . . 286

 Ejercicio 14-4: Niveles de dificultad. . . 286

Puntuación . . . 286

 Visualización de la partitura. . . 286

 Hacer un marcador. . . 287

 Actualización de la puntuación a medida que los extraterrestres son derribados. . . 289

 Restablecer la puntuación. . . 289

 Asegurándose de anotar todos los hits. . . 290

 Valores de puntos crecientes. . . 290

 Redondeando la puntuación. . . 291

 Puntajes altos. . . 292

 Mostrando el nivel. . . 294

 Visualización del número de barcos. . . 296

 Ejercicio 14-5: Puntuación más alta de todos los tiempos. . . 299

 Ejercicio 14-6: Refactorización. . . 299

 Ejercicio 14-7: Ampliando el juego. . . 299

 Ejercicio 14-8: Tirador lateral, versión final. . . 299

Resumen . . . 299

15

GENERANDO DATOS

301

Instalando Matplotlib. . . 302

Trazar un gráfico lineal simple. . . 302

 Cambiar el tipo de etiqueta y el grosor de la línea. . . 303

 Corrección de la trama. . . 305

 Uso de estilos integrados. . . 306

 Trazar y diseñar puntos individuales con scatter(). . . 306

 Trazar una serie de puntos con scatter(). . . 308

 Calcular datos automáticamente. . . 308

 Personalización de etiquetas de ticks. . . 309

 Definición de colores personalizados. . . 310

 Usando un mapa de colores. . . 310

 Guardar sus trazados automáticamente. . . 311

 Ejercicio 15-1: Cubos. . . 311

 Ejercicio 15-2: Cubos de colores. . . 311

Paseos aleatorios. . . 312

 Creando la clase RandomWalk. . . 312

 Elegir direcciones. . . 312

 Trazando el paseo aleatorio. . . 313

 Generando múltiples paseos aleatorios. . . 314

 Diseñando el paseo. . . 315

 Ejercicio 15-3: Movimiento molecular. . . 319

 Ejercicio 15-4: Paseos aleatorios modificados. . . 319

 Ejercicio 15-5: Refactorización. . . 319

Tirar dados con Plotly. . . 319

 Instalación de Plotly. . . 320

 Creando la clase de troquel. . . 320

 Tirando el dado. . . 320

 Analizando los resultados. . . 321

Hacer un histograma.....	322
Personalizando la trama. . .	323
Lanzar dos dados. . .	324
Más personalizaciones. . .	325
Dados rodantes de diferentes tamaños. . .	326
Ahorro de figuras. . .	327
Ejercicio 15-6: dos D8. . .	328
Ejercicio 15-7: Tres dados. . .	328
Ejercicio 15-8: Multiplicación de dados. . .	328
Ejercicio 15-9: Comprensiones de datos. . .	328
Ejercicio 15-10: practicando con ambas bibliotecas. . .	328
Resumen . .	328
<small>dieciséis</small>	
DESCARGAR DATOS	329
El formato de archivo CSV. . .	330
Analizando los encabezados del archivo CSV. . .	330
Impresión de los encabezados y sus posiciones. . .	331
Extracción y lectura de datos. . .	332
Trazar datos en un gráfico de temperatura. . .	332
El módulo de fecha y hora.	333
Trazar fechas.....	334
Trazar un periodo de tiempo más largo: . . .	336
Trazar una segunda serie de datos. . .	336
Sombrear un área en el gráfico. . .	337
Comprobación de errores . .	338
Descarga de sus propios datos. . .	341
Ejercicio 16-1: Lluvias de Sitka. . .	342
Ejercicio 16-2: Comparación entre Sitka y el Valle de la Muerte. . .	342
Ejercicio 16-3: San Francisco. . .	342
Ejercicio 16-4: Índices automáticos. . .	342
Ejercicio 16-5: Explorar. . .	342
Mapeo de conjuntos de datos globales: formato GeoJSON. . .	342
Descarga de datos sobre terremotos. . .	343
Examinando datos GeoJSON. . .	343
Hacer una lista de todos los terremotos. ...	345
Extrayendo Magnitudes. . .	346
Extracción de datos de ubicación: . . .	346
Construyendo un mapa mundial. . .	347
Representando Magnitudes. . .	348
Personalización de los colores de los marcadores: . . .	349
Otras escalas de colores.....	350
Agregar texto flotante. . .	350
Ejercicio 16-6: Refactorización. . .	352
Ejercicio 16-7: Título automatizado. . .	352
Ejercicio 16-8: Terremotos recientes. . .	352
Ejercicio 16-9: Incendios mundiales. . .	352
Resumen . .	352

17

TRABAJANDO CON APIS 355

Usando una 355

 API Git y GitHub. 356

 Solicitar datos mediante una llamada API. 356

 Instalar solicitudes. 357

 Procesando una respuesta API. 357

 Trabajar con el diccionario de respuestas. 358

 Resumiendo los principales repositorios. 361

 Monitoreo de límites de tasa API. 362

Visualización de repositorios utilizando Plotly. 362

 Aplicar estilo al gráfico. 364

 Agregar información sobre herramientas personalizadas. 365

 Agregar enlaces en los que se puede hacer clic. 366

 Personalización de los colores de los marcadores. 367

 Más información sobre Plotly y la API de GitHub 368

La API de noticias para hackers 368

 Ejercicio 17-1: Otros idiomas. 371

 Ejercicio 17-2: Discusiones activas. 371

 Ejercicio 17-3: Prueba de python_repos.py. 372

 Ejercicio 17-4: Exploración adicional. 372

Resumen . . 372

18

COMENZANDO CON DJANGO 373

Configuración de un proyecto. : : 374

 Escribir una especificación.: 374

 Creando un entorno virtual. . 374

 Activando el Entorno Virtual. . 375

 Instalando Django. . . 375

 Creando un proyecto en Django. . . 376

 Creando la Base de Datos.: 376

 Visualizando el Proyecto.: 377

 Ejercicio 18-1: Nuevos proyectos. 378

Iniciar una aplicación. . 379

 Definición de modelos. : : 379 : : : : :

 Activando Modelos. . 380

 El sitio de administración de Django. : : 381

 Definición del modelo de entrada: : 384

 Migrando el modelo de entrada. : : 385

 Registro de entrada en el sitio de administración. : : 385

 El caparazón de Django. : : 386

 Ejercicio 18-2: Entradas breves. 387

 Ejercicio 18-3: La API de Django. 388

 Ejercicio 18-4: Pizzería. 388

Creación de páginas: la página de inicio del registro de aprendizaje. 388

 Mapeo de una URL. 388

 Escribir una vista. 390

 Escribir una plantilla. 390

Ejercicio 18-5: Planificador de comidas.	392
Ejercicio 18-6: Página de inicio de pizzería.	392
Construyendo páginas adicionales.	392
Herencia de plantilla.	392
La página de temas.	394
Páginas de temas individuales.	397
Ejercicio 18-7: Documentación de plantilla.	400
Ejercicio 18-8: Páginas de pizzería.	400
Resumen	400
19	
CUENTAS DE USUARIO	403
Permitir a los usuarios ingresar datos.	404
Agregar nuevos temas.	404
Agregar nuevas entradas.	408
Edición de entradas.	412
Ejercicio 19-1: Blog.	415
Configuración de cuentas de usuario.	415
La App de cuentas.	415
La página de inicio de sesión.	416
Saliendo de tu cuenta.	419
La página de registro.	420
Ejercicio 19-2: Cuentas de blog.	423
Permitir que los usuarios sean propietarios de sus datos.	423
Restringir el acceso con @login_required.	423
Conexión de datos a determinados usuarios.	425
Restringir el acceso a los temas a los usuarios adecuados.	427
Proteger los temas de un usuario.	428
Protegiendo la página edit_entry.	429
Asociación de nuevos temas con el usuario actual.	429
Ejercicio 19-3: Refactorización.	430
Ejercicio 19-4: Protección de nueva_entrada.	430
Ejercicio 19-5: Blog protegido.	430
Resumen	430
20	
DISEÑAR E IMPLEMENTAR UNA APLICACIÓN	433
Registro de aprendizaje de estilo.	434
La aplicación django-bootstrap5.	434
Uso de Bootstrap para diseñar el registro de aprendizaje.	434
Modificando base.html.	435
Diseñar la página de inicio usando un Jumbotron.	440
Aplicar estilo a la página de inicio de sesión.	441
Aplicar estilo a la página de temas.	442
Aplicar estilo a las entradas en la página del tema.	443
Ejercicio 20-1: Otras formas.	445
Ejercicio 20-2: Blog con estilo.	445
Implementación del registro de aprendizaje.	445
Crear una cuenta en Platform.sh.	445
Instalación de la CLI Platform.sh.	446
Instalando la plataformashconfig.	446

Creando un archivo de requisitos.txt.	446
Requisitos de implementación adicionales.	447
Agregar archivos de configuración.	447
Modificando settings.py para Platform.sh.	451
Usando Git para rastrear los archivos del proyecto.	451
Creando un proyecto en Platform.sh.	453
Empujando a Platform.sh.	455
Visualización del proyecto en vivo.	456
Refinando la implementación de Platform.sh.	456
Creación de páginas de error personalizadas.	459
Desarrollo en curso	460
Eliminar un proyecto en Platform.sh.	461
Ejercicio 20-3: Blog en vivo.	461
Ejercicio 20-4: Registro de aprendizaje ampliado.	462
Resumen	462
A	
INSTALACIÓN Y SOLUCIÓN DE PROBLEMAS Python en	463
Windows.	463
Usando py en lugar de python.	463
Volviendo a ejecutar el instalador.	464
Python en macOS	464
Instalar accidentalmente la versión de Python de Apple.	464
Python 2 en versiones anteriores de macOS.	465
Pitón en Linux.	465
Usando la instalación predeterminada de Python.	465
Instalación de la última versión de Python.	465
Comprobar qué versión de Python estás utilizando.	466
Palabras clave de Python y funciones integradas.	466
Palabras clave de Python.	466
Funciones integradas de Python.	467
B	
EDITORES DE TEXTO E IDEs que	469
funcionan de manera eficiente con VS Code.	470
Configurando el código VS.	470
Atajos de código VS.	473
Otros editores de texto e IDE.	474
INACTIVO	474
Geany.	474
Texto sublime	474
Emacs y Vim.	475
PyCharm.	475
Cuadernos Jupyter.	475
C	
OBTENER AYUDA	477
Primeros pasos.....	477
Pruébalo otra vez :	478
Tomar un descanso	478
Consulte los recursos de este libro.	478

Buscando en línea.	479
Desbordamiento de pila	479
La documentación oficial de Python.	479
Documentación Oficial de la	480
Biblioteca. r/learnpython:	480
Publicaciones de blog.	480
Discordia.	480
Flojo	481
D	
USANDO GIT PARA EL CONTROL DE VERSIONES	483
Instalando Git.	484
Configurando Git.	484
Elaboración de un proyecto.....	484
Ignorar archivos.	484
Inicializando un repositorio.	485
Comprobando el estado.	485
Agregar archivos al repositorio.	486
Hacer un compromiso.	486
Comprobando el registro.	487
El segundo compromiso.	487
Abandonar los cambios.	488
Consultando confirmaciones anteriores.	489
Eliminando el repositorio.	491
mi	
SOLUCIÓN DE PROBLEMAS DE IMPLEMENTACIONES	493
Comprensión de las implementaciones.	494
Solución de problemas básicos.	494
Siga las sugerencias en pantalla.	495
Lea la salida del registro.	496
Solución de problemas específicos del sistema operativo.	497
Implementación desde Windows.	497
Implementación desde macOS.	499
Implementación desde Linux.	499
Otros enfoques de implementación.	500
ÍNDICE	503

PREFACIO A LA TERCERA EDICIÓN

La respuesta a la primera y segunda edición del Python Crash Course ha sido abrumadoramente positiva. Se han impreso más de un millón de ejemplares, incluidas traducciones a más de 10 idiomas. He recibido cartas y correos electrónicos de lectores de tan sólo 10 años, así como de jubilados que quieren aprender a programar en su tiempo libre. Python Crash Course se utiliza en escuelas intermedias y secundarias, y también en clases universitarias. Los estudiantes a los que se les asignan libros de texto más avanzados utilizan Python Crash Course como texto complementario para sus clases y lo consideran un complemento valioso. La gente lo está utilizando para mejorar sus habilidades en el trabajo, cambiar de carrera y comenzar a trabajar en sus propios proyectos paralelos. En resumen, la gente está utilizando el libro para toda la gama de propósitos que esperaba, y mucho más.

Ha sido un placer tener la oportunidad de escribir una tercera edición del curso intensivo de Python . Aunque Python es un lenguaje maduro, continúa evolucionando como lo hacen todos los lenguajes. Mi objetivo principal al revisar el libro es mantenerlo como un curso introductorio de Python bien diseñado. Al leer este libro, aprenderá todo lo que necesita para comenzar a trabajar en sus propios proyectos y también construirá una base sólida para todo su aprendizaje futuro. Actualicé algunas secciones para reflejar formas más nuevas y simples de hacer las cosas en Python. También aclaré algunas secciones en las que ciertos detalles del lenguaje no se presentaban con la precisión que podrían haber sido. Todos los proyectos se han actualizado completamente utilizando bibliotecas populares y bien mantenidas que puede utilizar con confianza para crear sus propios proyectos.

El siguiente es un resumen de los cambios específicos que se han realizado en la tercera edición:

- El Capítulo 1 ahora presenta el editor de texto VS Code, que es popular entre programadores principiantes y profesionales y funciona bien en todos los sistemas operativos.
- El Capítulo 2 incluye los nuevos métodos `removeprefix()` y `removesuffix()`, que son útiles cuando se trabaja con archivos y URL. Este capítulo también presenta los mensajes de error recientemente mejorados de Python, que brindan información mucho más específica para ayudarlo a solucionar problemas de su código cuando algo sale mal.
- El Capítulo 10 utiliza el módulo `pathlib` para trabajar con archivos. Este es un enfoque mucho más sencillo para leer y escribir en archivos.
- El Capítulo 11 utiliza `pytest` para escribir pruebas automatizadas para el código que escribe. La biblioteca `pytest` se ha convertido en la herramienta estándar de la industria para escribir pruebas en Python. Es lo suficientemente amigable como para usarlo en tus primeras pruebas y, si sigues una carrera como programador de Python, también lo usarás en entornos profesionales.
- El proyecto Alien Invasion en los Capítulos 12-14 incluye un escenario para controlar la velocidad de cuadros, lo que hace que el juego se ejecute de manera más consistente en diferentes sistemas operativos. Se utiliza un enfoque más simple para construir la flota de extraterrestres y también se ha limpiado la organización general del proyecto.
- Los proyectos de visualización de datos de los Capítulos 15 a 17 utilizan las funciones más recientes de `Matplotlib` y `Plotly`. Las visualizaciones de `Matplotlib` presentan configuraciones de estilo actualizadas. El proyecto de caminata aleatoria tiene una pequeña mejora que aumenta la precisión de los gráficos, lo que significa que verá surgir una variedad más amplia de patrones cada vez que genere una nueva caminata. Todos los proyectos que presentan `Plotly` ahora utilizan el módulo `Plotly Express`, que le permite generar sus visualizaciones iniciales con solo unas pocas líneas de código. Puede explorar fácilmente una variedad de visualizaciones antes de comprometerse con un tipo de trama y luego concentrarse en refinar los elementos individuales de esa trama.
- El proyecto Registro de aprendizaje de los capítulos 18 a 20 se creó utilizando la última versión. Versión de Django y diseñada con la última versión de Bootstrap. Se ha cambiado el nombre de algunas partes del proyecto para que sea más fácil seguir la organización general del proyecto. El proyecto ahora está implementado en Platform.sh, un moderno servicio de alojamiento para proyectos de Django. El proceso de implementación está controlado por archivos de configuración YAML, que le brindan un gran control sobre cómo se implementa su proyecto. Este enfoque es consistente con la forma en que los programadores profesionales implementan proyectos modernos de Django.
- El Apéndice A se ha actualizado completamente para recomendar las mejores prácticas actuales para instalar Python en los principales sistemas operativos. El Apéndice B incluye instrucciones detalladas para configurar VS Code y breves descripciones de la mayoría de los principales editores de texto e IDE que se utilizan actualmente. Apéndice C dirige a los lectores a varios de los recursos en línea más populares para obtener

ayuda. El Apéndice D continúa ofreciendo un mini curso intensivo sobre el uso de Git para el control de versiones. El Apéndice E es completamente nuevo para la tercera edición. Incluso con un buen conjunto de instrucciones para implementar las aplicaciones que crea, hay muchas cosas que pueden salir mal. Este apéndice ofrece una guía detallada de solución de problemas que puede utilizar cuando el proceso de implementación no funciona en el primer intento.

- El índice se ha actualizado completamente para permitirle utilizar el Curso intensivo de Python como referencia para todos sus proyectos futuros de Python.

¡Gracias por leer el curso intensivo de Python! Si tienes algún comentario o preguntas, no dude en ponerse en contacto; Soy @ehmatthes en Twitter.

EXPRESIONES DE GRATITUD

Este libro no habría sido posible sin el maravilloso y extremadamente profesional personal de No Starch Press. Bill Pollock me invitó a escribir un libro introductorio y aprecio profundamente esa oferta original. Liz Chadwick ha trabajado en las tres ediciones y el libro es mejor gracias a su participación continua. Eva Morrow aportó nuevos ojos a esta nueva edición y sus ideas también han mejorado el libro. Aprecio la guía de Doug McNair sobre el uso de la gramática adecuada, sin volverse demasiado formal. Jennifer Kepler ha supervisado el trabajo de producción, que convierte mis

numerosos archivos en un producto final pulido.

Hay muchas personas en No Starch Press que han ayudado a que este libro sea un éxito pero con quienes no he tenido la oportunidad de trabajar directamente. No Starch cuenta con un equipo de marketing fantástico, que va más allá de la simple venta de libros; se aseguran de que los lectores encuentren los libros que probablemente les funcionen bien y les ayuden a alcanzar sus objetivos. No Starch también tiene un sólido departamento de derechos extranjeros. El curso intensivo de Python ha llegado a lectores de todo el mundo, en muchos idiomas, gracias a la diligencia de este equipo. A todas estas personas con las que no he trabajado individualmente, gracias por ayudar a Python Crash Course a encontrar su audiencia.

Me gustaría agradecer a Kenneth Love, el revisor técnico de los tres ediciones del curso intensivo de Python. Conocí a Kenneth en PyCon un año y su entusiasmo por el lenguaje y la comunidad Python ha sido una fuente constante de inspiración profesional desde entonces. Kenneth, como siempre, fue más allá de la simple verificación de hechos y revisó el libro con el objetivo de

Ayudar a los programadores más nuevos a desarrollar una comprensión sólida del lenguaje Python y la programación en general. También estuvieron atentos a las áreas que funcionaron bastante bien en ediciones anteriores pero que podrían mejorarse si se les diera la oportunidad de realizar una reescritura completa. Dicho esto, cualquier inexactitud que quede es completamente mía.

También me gustaría expresar mi agradecimiento a todos los lectores que han compartido su experiencia trabajando a través del Python Crash Course. Aprender los conceptos básicos de la programación puede cambiar tu perspectiva sobre el mundo y, a veces, esto tiene un profundo impacto en las personas. Es una profunda lección de humildad escuchar estas historias y apreciar a todos los que han compartido sus experiencias tan abiertamente.

Me gustaría agradecer a mi padre por iniciarme en la programación desde muy joven y por no tener miedo de que rompiera su equipo. Me gustaría agradecer a mi esposa, Erin, por apoyarme y alentarme durante la escritura de este libro y durante todo el trabajo necesario para mantenerlo en múltiples ediciones. También me gustaría agradecer a mi hijo Ever, cuya curiosidad continúa inspirándome.

INTRODUCCIÓN



Cada programador tiene una historia sobre cómo aprendió a escribir su primer programa. Comencé a programar cuando era niño, cuando

mi padre trabajaba para Digital Equipment Corporation, una de las empresas pioneras de la era informática moderna. Escribí mi primer programa en un kit de computadora que mi papá había ensamblado en nuestro sótano. La computadora no consistía más que en una placa base conectada a un teclado sin carcasa, y su monitor era un tubo de rayos catódicos desnudo. Mi programa inicial era un simple juego de adivinanzas de números, que se parecía a esto:

¡Estoy pensando en un número! Intenta adivinar el número en el que estoy pensando: 25

¡Demasiado baja! Adivina de nuevo: 50

¡Demasiado alto! Adivina de nuevo: 42

¡Eso es todo! ¿Te gustaría volver a jugar? (sí/no) no

¡Gracias por jugar!

Siempre recordaré lo satisfecho que me sentí al ver a mi familia jugar un juego que creé y que funcionó como esperaba.

Esa primera experiencia tuvo un impacto duradero. Hay una verdadera satisfacción en construir algo con un propósito, que resuelva un problema. El software que escribo ahora satisface necesidades más importantes que los esfuerzos de mi infancia, pero la sensación de satisfacción que obtengo al crear un programa que funciona sigue siendo en gran medida la misma.

¿Para quién es este libro?

El objetivo de este libro es ponerlo al día con Python lo más rápido posible para que pueda crear programas que funcionen (juegos, visualizaciones de datos y aplicaciones web) mientras desarrolla una base en programación que le será útil para el resto de su vida. su vida. El curso intensivo de Python está escrito para personas de cualquier edad que nunca han programado en Python o nunca han programado nada. Este libro es para aquellos que desean aprender rápidamente los conceptos básicos de la programación para poder concentrarse en proyectos interesantes y para aquellos a quienes les gusta poner a prueba su comprensión de nuevos conceptos resolviendo problemas significativos. Python Crash Course también es perfecto para profesores de todos los niveles que quieran ofrecer a sus alumnos una introducción a la programación basada en proyectos. Si estás tomando una clase universitaria y quieres una introducción a Python más amigable que el texto que te han asignado, este libro también puede hacer que tu clase sea más fácil. Si está buscando cambiar de carrera, Python Crash Course puede ayudarlo a realizar la transición hacia una carrera profesional más satisfactoria. Ha funcionado bien para una amplia variedad de lectores, con una amplia gama de objetivos.

¿Qué puedes esperar aprender?

El propósito de este libro es convertirte en un buen programador en general y en un buen programador de Python en particular. Aprenderá de manera eficiente y adoptará buenos hábitos a medida que obtenga una base sólida en conceptos generales de programación. Después de avanzar en el curso intensivo de Python, debería estar listo para pasar a técnicas de Python más avanzadas y su próximo lenguaje de programación será aún más fácil de comprender.

En la Parte I de este libro, aprenderá los conceptos básicos de programación que necesita para escribir programas en Python. Estos conceptos son los mismos que aprendería al comenzar en casi cualquier lenguaje de programación. Aprenderá sobre diferentes tipos de datos y las formas en que puede almacenar datos en sus programas. Creará colecciones de datos, como listas y diccionarios, y trabajará con esas colecciones de manera eficiente. Aprenderá a usar bucles while y sentencias if para probar ciertas condiciones, de modo que pueda ejecutar secciones específicas de código mientras esas condiciones son verdaderas y ejecutar otras secciones cuando no lo son, una técnica que le ayuda a automatizar muchos procesos.

Aprenderá a aceptar aportaciones de los usuarios para que sus programas sean interactivos y a mantenerlos ejecutándose todo el tiempo que el usuario desee. Explorarás cómo escribir funciones que hagan que partes de tu programa sean reutilizables.

por lo que solo tienes que escribir bloques de código que realicen ciertas acciones una vez, mientras usas ese código tantas veces como necesites. Luego ampliará este concepto a comportamientos más complicados con clases, haciendo que programas bastante simples respondan a una variedad de situaciones. Aprenderá a escribir programas que manejen correctamente los errores comunes. Después de trabajar en cada uno de estos conceptos básicos, escribirá una serie de programas cada vez más complejos utilizando lo que ha aprendido. Finalmente, darás el primer paso hacia la programación intermedia aprendiendo a escribir pruebas para tu código, de modo que puedas desarrollar más tus programas sin preocuparte por introducir errores.

Toda la información de la Parte I lo preparará para asumir proyectos más grandes y complejos.

En la Parte II, aplicará lo aprendido en la Parte I a tres proyectos. Puede realizar cualquiera o todos estos proyectos, en el orden que mejor le convenga. En el primer proyecto, en los capítulos 12 a 14, crearás un juego de disparos al estilo de Space Invaders llamado Alien Invasion, que incluye varios niveles de juego cada vez más difíciles. Una vez que hayas completado este proyecto, deberías estar en camino de poder desarrollar tus propios juegos 2D. Incluso si no aspiras a convertirte en programador de juegos, trabajar en este proyecto es una forma divertida de unir gran parte de lo que aprenderás en la Parte I.

El segundo proyecto, en los capítulos 15 a 17, le presenta la visualización de datos.

Los científicos de datos utilizan una variedad de técnicas de visualización para ayudar a dar sentido a la gran cantidad de información disponible. Trabaja con conjuntos de datos que genere mediante código, conjuntos de datos que descargue de fuentes en línea y conjuntos de datos que sus programas descarguen automáticamente. Una vez que haya completado este proyecto, podrá escribir programas que analicen grandes conjuntos de datos y creen representaciones visuales de muchos tipos diferentes de información.

En el tercer proyecto, en los Capítulos 18 y 20, creará una pequeña aplicación web llamada Registro de aprendizaje. Este proyecto le permite llevar un diario organizado de la información que ha aprendido sobre un tema específico. Podrá mantener registros separados para diferentes temas y permitir que otros creen una cuenta y comiencen sus propios diarios. También aprenderá cómo implementar su proyecto para que cualquiera pueda acceder a él en línea, desde cualquier parte del mundo.

Recursos en línea

No Starch Press tiene más información sobre este libro disponible en línea en <https://nostarch.com/python-crash-course-3rd-edition>.

También mantengo un amplio conjunto de recursos complementarios en https://ehmatthes.github.io/pcc_3e. Estos recursos incluyen lo siguiente:

Instrucciones de configuración Las instrucciones de configuración en línea son idénticas a las del libro, pero incluyen enlaces activos en los que puede hacer clic para seguir los diferentes pasos. Si tiene algún problema de configuración, consulte este recurso.

Actualizaciones Python como todos los lenguajes, está en constante evolución. Mantengo un conjunto completo de actualizaciones, por lo que si algo no funciona, consulte aquí para ver si las instrucciones han cambiado.

Soluciones a los ejercicios Debe dedicar mucho tiempo a realizar los ejercicios de las secciones “Pruébelo usted mismo”. Sin embargo, si está estancado y no puede progresar, las soluciones para la mayoría de los ejercicios están en línea.

Hojas de referencia También está disponible en línea un conjunto completo de hojas de referencia descargables para una referencia rápida a los conceptos principales.

¿Por qué Python?

Cada año, considero si debo seguir usando Python o pasar a un lenguaje diferente, quizás uno que sea más nuevo en el mundo de la programación.

Pero sigo centrándome en Python por muchas razones. Python es un lenguaje increíblemente eficiente: sus programas harán más en menos líneas de código de lo que requerirían muchos otros lenguajes. La sintaxis de Python también te ayudará a escribir código "limpio". Su código será más fácil de leer, más fácil de depurar y más fácil de ampliar y desarrollar, en comparación con otros lenguajes.

La gente usa Python para muchos propósitos: crear juegos, crear aplicaciones web, resolver problemas comerciales y desarrollar herramientas internas en todo tipo de empresas interesantes. Python también se utiliza mucho en campos científicos, para investigación académica y trabajos aplicados.

Una de las razones más importantes por las que sigo usando Python es porque de la comunidad Python, que incluye un grupo de personas increíblemente diverso y acogedor. La comunidad es esencial para los programadores porque la programación no es una actividad solitaria. La mayoría de nosotros, incluso los programadores más experimentados, necesitamos pedir consejo a otras personas que ya han resuelto problemas similares. Tener una comunidad bien conectada y de apoyo es fundamental para ayudarle a resolver problemas, y la comunidad de Python apoya plenamente a las personas que están aprendiendo Python como su primer lenguaje de programación o que llegan a Python con experiencia en otros lenguajes.

Python es un gran lenguaje para aprender, ¡así que comencemos!

PARTE I

LO ESENCIAL

La Parte I de este libro le enseña los conceptos básicos que necesitará para escribir programas en Python. Muchos de estos conceptos son comunes a todos los lenguajes de programación, por lo que te serán útiles a lo largo de tu vida como programador.

En el Capítulo 1 instalarás Python en tu computadora y ejecutarás tu primer programa, que imprime el mensaje ¡Hola mundo! a la pantalla.

En el Capítulo 2 aprenderá a asignar información a variables y a trabajar con texto y valores numéricos.

Los capítulos 3 y 4 presentan listas. Las listas pueden almacenar tanta información como desee en un solo lugar, lo que le permitirá trabajar con esos datos de manera eficiente. Podrás trabajar con cientos, miles e incluso millones de valores en tan solo unas pocas líneas de código.

En el Capítulo 5 usará sentencias if para escribir código que responda de una manera si ciertas condiciones son verdaderas y responda de otra manera si esas condiciones no son verdaderas.

El Capítulo 6 le muestra cómo utilizar los diccionarios de Python, que le permiten establecer conexiones entre diferentes piezas de información. Al igual que las listas, los diccionarios pueden contener tanta información como necesite almacenar.

En el Capítulo 7 aprenderá cómo aceptar entradas de los usuarios para que sus programas sean interactivos. También aprenderá sobre los bucles while, que ejecutan bloques de código repetidamente siempre que ciertas condiciones sigan siendo verdaderas.

En el Capítulo 8 escribirás funciones, que se denominan bloques de código que realizan una tarea específica y se pueden ejecutar cuando los necesite.

El Capítulo 9 presenta clases que le permiten modelar objetos del mundo real.

Escribirás código que represente perros, gatos, personas, automóviles, cohetes y más.

El Capítulo 10 le muestra cómo trabajar con archivos y manejar errores para que sus programas no colapsen inesperadamente. Almacenará datos antes de que se cierre el programa y los volverá a leer cuando el programa se ejecute nuevamente. Aprenderá sobre las excepciones de Python, que le permiten anticipar errores y hacer que sus programas los manejen correctamente.

En el Capítulo 11 aprenderá a escribir pruebas para su código, para comprobar que sus programas funcionan de la forma prevista. Como resultado, podrá ampliar sus programas sin preocuparse por introducir nuevos errores.

Probar tu código es una de las primeras habilidades que te ayudarán a pasar de programador principiante a programador intermedio.