Iniciando a Programar con Python

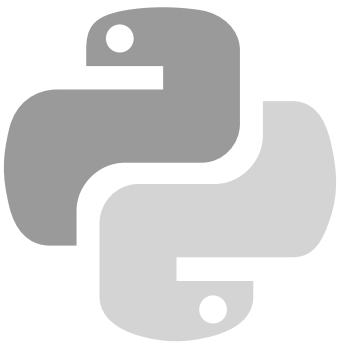
Guía básica de programación

Javier Cuervo Álvarez
Lina María Cuervo Díaz
Nathalia Andrea Cuervo Díaz

FORMACIÓN EDUCACIÓN

EDITORIAL UPTC

Iniciando a Programar con **Python**



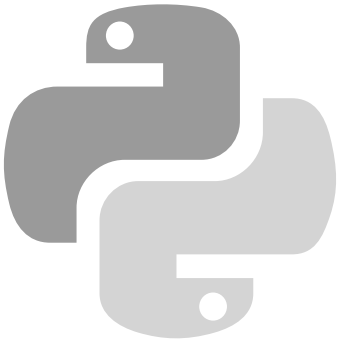
Guía básica de programación

Javier Cuervo Álvarez
Lina María Cuervo Díaz
Nathalia Andrea Cuervo Díaz

Uptc®
Universidad Pedagógica y Tecnológica de Colombia
VIGILADA MINEDUCACIÓN

EDITORIAL UPTC

Iniciando a Programar con **Python**



Guía básica de programación

Javier Cuervo Álvarez
Lina María Cuervo Díaz
Nathalia Andrea Cuervo Díaz

ENFOQUE: FORMACIÓN
COLECCIÓN: EDUCACIÓN
Colección UPTC N° 87

Iniciando a Programar con Python. Guía básica de programación
Beginning Programming with Python: Basic Programming Guide

Primera Edición, 2024

ISBN (POD) 978-958-660-903-6
ISBN (ePub) 978-958-660-900-5

Impreso y hecho en Colombia - Printed and made in Colombia

Iniciando a Programar con Python. Guía básica de programación / Beginning Programming with Python: Basic Programming Guide. / Cuervo Álvarez, Javier. Tunja: Editorial UPTC, 2024. 346 p.

ISBN (POD) 978-958-660-903-6
ISBN (ePub) 978-958-660-900-5

Incluye referencias bibliográficas.

1. Python. 2. Programación orientada a objetos. 3. Visual Studio Code. 4. Jupyter Notebook. 5. Estructuras de datos. 6. Estructura de control. (Dewey 005/21) (Thema UM-Programación informática/desarrollo de software)

Editor
Óscar Pulido Cortés

Corrección de Estilo
Nelson Zorro Gutiérrez

Impresión
Búhos Editores Ltda.
Calle 57 No. 9 – 36
Cel. 314 411 5024
Tunja – Boyacá – Colombia

Libro de formación y/o académico.

Citar este libro / Cite this book
Cuervo Álvarez, J. (2024). *Iniciando a Programar con Python. Guía básica de programación*. Editorial UPTC.. Editorial UPTC.
doi.org/10.19053/uptc.9789586609036

Editorial UPTC
La Colina, Bloque 7, Casa 5
Avenida Central del Norte No. 39-115, Tunja, Boyacá
comite.editorial@uptc.edu.co
www.uptc.edu.co
https://editorial.uptc.edu.co

RESUMEN

"Iniciando a Programar con Python: Guía básica de programación" ofrece una introducción completa al lenguaje de programación Python. Comienza con una visión histórica y las ventajas de Python, para luego sumergirse en el proceso de instalación, que incluye Jupyter y Visual Studio Code. Cubre conceptos fundamentales de programación como identificadores, palabras clave, bloques de código, entrada/salida y comentarios. La sección sobre variables, constantes y tipos de datos explora números enteros, de punto flotante, booleanos, variables de texto, listas, conjuntos, tuplas y diccionarios. Presenta operadores, desde básicos hasta bit a bit.

Luego se abordan las estructuras de control, funciones y módulos, con la exploración de temas como manejo de excepciones y bucles. También se discuten conceptos avanzados, como la programación orientada a objetos, matrices y variables globales/locales. El libro concluye con una exploración extensa de estructuras de datos, la cual cubre listas, conjuntos, tuplas y diccionarios en términos de características, operaciones y métodos. La última sección aborda la entrada/salida con archivos y abarca archivos de texto y binarios, manejo de archivos CSV y serialización de datos. Esta guía proporciona una base sólida para principiantes en Python. Va desde conceptos básicos hasta temas avanzados de programación y estructuras de datos.

Palabras clave: Python; Programación orientada a objetos; Visual Studio Code; Jupyter Notebook; Estructuras de datos; Estructura de control.

ABSTRACT

"Beginning Programming with Python: Basic Programming Guide" offers a comprehensive introduction to the Python programming language. Beginning with a historical overview and the advantages of Python, the book dives into the installation process, including Jupyter and Visual Studio Code. It covers fundamental programming concepts such as identifiers, keywords, code blocks, input/output, and comments. The section on variables, constants, and data types explores integers numbers, floating-point, booleans, text variables, lists, sets, tuples, and dictionaries, presenting operators from basic to bitwise.

Subsequently, control structures, functions, and modules are addressed, exploring topics such as exception handling and loops. Advanced concepts like object-oriented programming, matrices, and global/local variables are also discussed. The book concludes with an extensive exploration of data structures, covering lists, sets, tuples, and dictionaries in terms of features, operations, and methods. The final section addresses file input/output, covering text and binary files, CSV file handling, and data serialization. This guide provides a solid foundation for beginners in Python, covering basic concepts and advancing to more advanced programming and data structure topics.

Keywords: Python; Object Oriented Programming; Visual Studio Code; Jupyter Notebook; Data Structures; Control Structure.

Contenido

Prefacio 17

Capítulo 1. Introducción | 21

Reseña histórica de Python 23
Ventajas de trabajar con Python 25
Acerca del libro 27
Convenciones del libro 27

Capítulo 2. Instalación de Python | 29

Ejecución de Python 33
Jupyter 35
Instalar Jupyter Notebook en Anaconda 36
- Instalación 36
- Guía básica de Jupyter con Anaconda 38
- Menú File (archivo) 40
- Menú Edit (editar) 41
- Menú View (ver) 42
- Menú Insert (insertar) 42
- Menú Cell (celda) 43
- Menú Kernel 44
- Menú Help (ayuda) 44
- Barra de herramientas 45
- Crear y guardar archivos 45

Visual Studio Code 49
- Instalación de Visual Studio Code 50
- Instalación de Jupyter en Visual Studio Code 52
- Entorno de Visual Studio Code 55
- Interface de usuario 56
- Diseño básico 57
- Barra de menú 58
- Guía básica de Jupyter con Visual Studio Code 70

Capítulo 3. Conceptos básicos | 83

Identificadores 85
Palabras clave reservadas 87
Bloques de código 88
Indentación 88
Suites 89
Entrada y salida de datos 91
Comentarios 91
    Comentario de una sola línea 92
    Comentarios de varias líneas 93
Sentencias multilínea 94
Múltiples declaraciones en una sola línea 95

Capítulo 4. Variables y constantes | 97

Tipos de datos 100
Sintaxis para definir variables 101
Variables enteras 102
Variables reales o de punto flotante 104
Variables booleanas 105
Variables de texto 105
Listas 106
Conjuntos 106
Tuplas 106
Diccionarios 107
If en línea 108
Constantes 108
Conversión de tipos de datos 109

Capítulo 5. Operadores | 111

Operadores básicos 113
Operadores de asignación 113
Operadores aritméticos 115
Operadores de comparación o relacionales 117
Operadores lógicos 119
Operadores de membresía 121
Operadores de identidad 122
Operadores bit a bit 123
Operadores de secuencia 125
Expresiones compuestas y precedencia de operadores 125

Capítulo 6. Estructuras de control | 127

Selección "if" 129
Selección "if-else" 130
Selección anidada "if-elif-else" 132
Selección anidada "if-esle" 134
Estructuras de Iteración 136
Estructura "for" 136
Bucles "for" anidados 139
Instrucción "break" 141
Instrucción "continue" 142
Estructura "while" 143
Bucles "while" anidados 146
Excepciones 147
Bloques "try-except" 148

Capítulo 7. Funciones y módulos | 151

Definición de funciones 153
Invocar funciones 154
Módulos 155
Importar todo el módulo 155
Importar una función específica del módulo 155
Importar todo el módulo con un alias 156
Ejemplo de importación de módulos 156
Funciones con parámetros 159
Función para comprobar si un número es primo 160
Invocar funciones 161
Variables globales y locales 164
Valores de retorno 165
Ejemplo de aplicación 166
Arreglos 166
Funciones para el manejo de matrices 167
Función para visualizar una matriz 168
Suma de matrices 169
Resta de matrices 171
Multiplicación de matrices 172
Matriz transpuesta 175
Matriz inversa 178
Solución de sistema de ecuaciones 181

Capítulo 8. Programación orientada a objetos | 187

Objetos, clases e instancias 190
Clases 192
Instancias 196
Atributo de instancia dinámico 198
Herencia 199
Herencia múltiple 206
Herencia de clases integradas 210
Polimorfismo 212
Clases abstractas 214
Sobrecarga de operadores 217
Ejemplo de aplicación 220

Capítulo 9. Estructuras de datos | 231

Listas 233
- Características 234
- Operaciones y métodos 234
- Crear listas y agregar elementos 237
- Ordenación de listas 238
- Insertar elementos 240
- Eliminar y retornar elementos de una lista 241
- Eliminar elementos de la lista 243
- Número de elementos de una lista 244
- Tipos de elementos en una lista 245
- Agregar los elementos de otra lista 245

Conjuntos 246
- Características 247
- Operaciones y métodos 247
- Definición 249
- Unión de conjuntos 250
- Intersección 251
- Diferencia de conjuntos 253
- Diferencia simétrica de conjuntos 254
- Operaciones básicas 255

Tuplas 257
- Características 257
- Operaciones y métodos 257
- Definición de una tupla 259

Diccionarios 261
- Características 261
- Operaciones y métodos 261
- Casos de uso de diccionarios 263

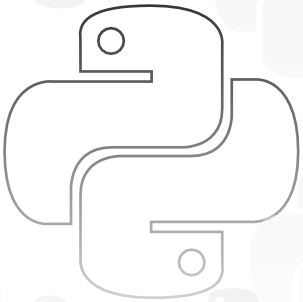
Definición de diccionarios 264
Acceder a elementos 267
Agregar un nuevo elemento 268
Eliminar elementos 269
Mostrar datos del diccionario completo 270
Operaciones básicas 271

Capítulo 10. Entrada y salida con archivos | 275

Archivos tipo texto 278
Abrir y cerrar archivos 279
Cierre de archivos 281
Lectura de archivos 281
Escritura en archivos 284
Lectura y escritura en archivos 285
Interacción con directorios y archivos 287
Archivos CSV 288
Lectura de archivos CSV 289
Escritura en archivos CSV 292
Lectura y escritura en archivos CSV 295
Lectura de datos de archivos CSV en diccionarios 297
Escritura de datos de diccionarios en archivos CSV 301
Actualización de datos de diccionarios en archivos CSV 306
Archivos binarios 312
Serialización de datos 312
Escritura datos serializados 313
Lectura datos serializados 316
Actualización de registros 319
Datos no serializados 321
Escritura datos no serializados 323
Lectura de datos no serializados 324
Buscar registros 327
Modificar registros 330

Índice 333

Bibliografía 343



Prefacio

Python es un lenguaje de programación de código abierto muy popular, ampliamente utilizado en la creación de programas y scripts en diversos campos. Es conocido por ser portátil, potente y relativamente fácil de aprender, lo cual lo convierte en una opción muy atractiva. Los programadores de todos los niveles de la industria del software han descubierto que el enfoque de Python en la productividad y en la calidad del software es una ventaja estratégica en proyectos de cualquier tamaño.

Ya sea si se está iniciando en la programación o si se es un desarrollador profesional, este libro está diseñado para ayudar a los lectores a familiarizarse con Python y proporcionarles un conocimiento más completo que otros recursos. Después de leer este libro, los lectores adquirirán suficiente conocimiento sobre Python para aplicarlo en cualquier área de interés que deseen explorar.

El libro se centra principalmente en los fundamentos del lenguaje Python, en lugar de sus aplicaciones específicas. Se concibe como un libro que busca establecer una base sólida para los lectores. Así mismo, se considera el primer volumen de una serie de libros sobre lenguajes de programación que tratan temas desde lo básico hasta lo avanzado y lo especializado en áreas de ciencias e ingenierías.

El objetivo del libro es ser utilizado en cursos introductorios de programación y como herramienta para aquellos que deseen aprender a programar de forma autodidacta este lenguaje. No se requieren conocimientos previos más allá de una comprensión básica de conceptos y de programas informáticos. El libro ofrece

contenido suficiente para un curso de programación a nivel universitario en cualquier área del conocimiento que lo requiera. Sus diez capítulos pueden ser utilizados como material en un curso universitario de un semestre.

Esta versión del libro cubre “Python 3.11.0” y utiliza ejemplos con características presentes en dicha versión. El sitio web principal del libro es: https://github.com/codigolibros/Iniciando-a-Programar-Con-Python, donde pueden descargar códigos fuente y los códigos en Jupyter Notebook.

En cuanto al estilo, se centra en la explicación de conceptos y en el desarrollo de ejemplos, similar a una clase teórico-práctica de programación, en lugar de seguir una estructura de libro de texto tradicional. Al final de cada ejemplo, se hace una explicación del código, como lo haría un profesor cuando está desarrollando su clase: se aborda un tema y se implementa el código respectivo. El libro no pretende ser una referencia exhaustiva de Python. Está dirigido principalmente a personas que están aprendiendo a programar por primera vez, por lo que abarca tanto conceptos generales de programación como aspectos específicos y prácticos de Python.

Cada capítulo cubre aspectos fundamentales de Python. En el capítulo 1 hay una breve reseña histórica de Python y se exploran las ventajas de trabajar con este lenguaje. También se proporciona una descripción general del libro y se establecen las convenciones que se utilizarán a lo largo del texto.

En el capítulo 2 se profundiza en la instalación de Python y se describen diferentes opciones, incluyendo el entorno Jupyter en el popular entorno de desarrollo integrado Visual Studio Code y Anaconda. Se aprende cómo instalar estas herramientas y cómo utilizarlas eficazmente para desarrollar programas en Python.

A partir del capítulo 3, se profundiza en los conceptos básicos de programación con Python. Se cubren temas como identificadores, palabras clave reservadas, bloques de código, entrada y salida de datos, comentarios y conversiones de tipos de datos. Estos

fundamentos son esenciales para comprender la sintaxis y la estructura de los programas en Python.

En el capítulo 4 se exploran en profundidad las variables y las constantes en Python. Se aprende sobre los diferentes tipos de datos, como números enteros, de punto flotante, booleanos y cadenas de texto. También se exploran estructuras de datos como listas, conjuntos, tuplas, diccionarios y cómo se trabaja con ellos. Asimismo, se discute la conversión de tipos de datos y cómo manipularlos según las necesidades.

El capítulo 5 se sumerge en los operadores en Python. Se exploran los operadores básicos, operadores de asignación, operadores aritméticos, operadores de comparación, operadores lógicos, operadores de membresía, operadores de identidad y operadores de secuencia. También se discute la precedencia de los operadores y cómo se utilizan las expresiones compuestas para realizar cálculos más complejos.

El capítulo 6 se centra en la estructura de control en Python. Se explican las estructuras de selección, "**if**", "**if-else**" e "**if-elif-else**" y cómo se utilizan para tomar decisiones en los programas. También se exploran las estructuras de iteración, como "**for**" y "**while**", y cómo se utilizan bucles para repetir tareas. Igualmente, se discuten las excepciones y cómo manejar errores y excepciones en los programas.

El capítulo 7 se adentra en las funciones y los módulos en Python. Se explica cómo definir y llamar funciones y cómo organizar el código en módulos reutilizables. También se explora el alcance de las variables. Así mismo, se discute la importación de módulos y las funcionalidades que ofrecen.

En el capítulo 8 se profundiza en la programación orientada a objetos en Python. Se analizan los conceptos de *objetos*, *clases* e *instancias* y se explora cómo emplear la orientación a objetos para estructurar el código y desarrollar programas con mayor flexibilidad y modularidad. Además, se abordan temas avanzados como la herencia, el polimorfismo y la sobrecarga de operadores, entre otros aspectos relevantes de la programación orientada a objetos.

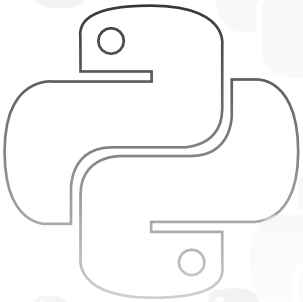
En el capítulo 9 se muestran las estructuras de datos en Python. Se profundiza sobre listas, conjuntos, tuplas y diccionarios, así como en su aplicación para almacenar y manipular información en los programas. Se exploran las operaciones y los métodos disponibles para cada tipo de estructura de datos y se discuten casos de uso prácticos.

Finalmente, el capítulo 10 se enfoca en la entrada y en la salida de archivos en Python. Se aprende sobre la forma abrir, cerrar, leer y escribir archivos de texto y se interactúa con directorios y archivos en el sistema. También se explora el manejo de archivos CSV (valores separados por comas) y archivos binarios, así como la serialización de datos.

Este libro ha sido escrito de manera clara y concisa, con ejemplos prácticos. No es necesario tener experiencia previa en programación, ya que se explica todo desde cero. Sin embargo, si los lectores ya tienen conocimientos básicos de programación, encontrarán este libro útil para ampliar sus habilidades y aplicarlas en Python. Así mismo, está diseñado para proporcionar a los lectores una introducción sólida al mundo de la programación en Python. Sin embargo, la naturaleza dinámica del desarrollo en Python y la amplia gama de bibliotecas disponibles hacen que sea esencial tener acceso a la documentación oficial, en: https://docs.python.org, para obtener información detallada sobre funciones específicas, módulos y prácticas recomendadas. Aconsejamos a los lectores que exploren este recurso invaluable para una comprensión más profunda y actualizada.

Capítulo 1

Introducción



Python es un lenguaje de programación de alto nivel, interpretado y de propósito general que ha ganado una gran popularidad en los últimos años. Es conocido por su sintaxis sencilla y por su legibilidad, lo que facilita su aprendizaje y su comprensión.

Python fue creado con el objetivo de ser un lenguaje fácil de usar y de entender. Guido Van Rossum, su creador, diseñó Python con una sintaxis clara y concisa. El enfoque de Python se centra en la legibilidad del código, lo que hace que sea más fácil para los programadores expresar sus ideas de manera efectiva.

Python también es un lenguaje multiplataforma, por lo cual puede ejecutarse en diferentes sistemas operativos, como Windows, macOS y Linux, sin necesidad de realizar grandes modificaciones en el código. Esto brinda flexibilidad a los desarrolladores y facilita la portabilidad de las aplicaciones.

Además, Python cuenta con una comunidad de desarrolladores muy activa y solidaria. Existen numerosos recursos en línea, tutoriales, documentación y foros de discusión donde los programadores pueden buscar ayuda, compartir conocimientos y colaborar en proyectos.

Reseña histórica de Python

Python fue creado a finales de la década de 1980, desarrollado en el lenguaje C por Guido Van Rossum, un programador holandés, quien comenzó a desarrollar Python en diciembre de 1989 como un proyecto de código abierto para su uso interno en el Centro de Matemáticas y Ciencias de la Computación (CWI, por sus siglas en neerlandés) en los Países Bajos.

La motivación detrás del desarrollo de Python era crear un lenguaje de programación que fuera fácil de leer, con una sintaxis clara y concisa, para permitir a los programadores expresar ideas de manera más efectiva. Van Rossum se basó en otros lenguajes existentes, como ABC y Modula-3, para desarrollar la sintaxis y las características de Python.

En febrero de 1991, Van Rossum publicó la primera versión de Python, conocida como Python 0.9.0. A medida que el lenguaje ganaba popularidad, se formó una comunidad de desarrolladores en línea que contribuyeron al desarrollo y a la mejora continua de Python.

En 2000 se lanzó Python 2.0, que introdujo características importantes, como la recolección de basura mejorada y soporte para Unicode. Python 2.0 se convirtió en la versión principal del lenguaje durante muchos años, pero en 2008 se anunció que Python 3.0 estaba en desarrollo.

Python 3.0 se lanzó en diciembre de 2008 y presentó cambios significativos en el lenguaje para mejorar la consistencia, la simplicidad y la eliminación de características obsoletas. Sin embargo, debido a algunas incompatibilidades con la versión anterior (Python 2.x), la adopción de Python 3 fue lenta al principio y muchas bibliotecas y aplicaciones existentes se basaron en Python 2.

En 2020 Python 2 alcanzó el final de su vida y ya no recibió actualizaciones o soporte oficial. Esto llevó a una migración más amplia hacia Python 3. La mayoría de las bibliotecas y frameworks importantes ahora son compatibles con Python 3.

A lo largo de los años, Python ha ganado una gran popularidad y se ha convertido en uno de los lenguajes de programación más utilizados en diversos campos como desarrollo web, análisis de datos, inteligencia artificial (IA), aprendizaje automático y ciencia de datos. La comunidad de Python ha crecido enormemente, con una amplia gama de bibliotecas y herramientas disponibles, lo que ha contribuido a su éxito y a su adopción generalizada en la industria de la tecnología.

Ventajas de trabajar con Python

Python es conocido por tener numerosas ventajas que lo han convertido en uno de los lenguajes de programación más populares y preferidos por los desarrolladores en diferentes campos.

Python se destaca por su sintaxis clara y legible. Su diseño se centra en la facilidad de comprensión del código, lo que permite a los programadores escribir programas de manera más intuitiva y comprensible. La simplicidad de su sintaxis hace que el código en Python sea más fácil de leer y de mantener, lo que ahorra tiempo y reduce los errores.

Python cuenta con una gran comunidad activa y solidaria de desarrolladores y entusiastas en todo el mundo, la cual proporciona una amplia variedad de recursos como bibliotecas, frameworks, tutoriales y documentación. Además, hay numerosos foros y grupos de discusión en línea donde los programadores pueden buscar ayuda, compartir conocimientos y colaborar en proyectos.

Python es un lenguaje multiplataforma que ofrece portabilidad y compatibilidad. Por tal razón, los programas escritos en Python pueden ser ejecutados en diferentes sistemas operativos, como Windows, macOS y Linux, sin requerir modificaciones significativas. Además, Python es compatible con otros lenguajes y se integra fácilmente con ellos, lo que permite a los desarrolladores aprovechar las fortalezas de diferentes herramientas y bibliotecas.

Python es considerado uno de los lenguajes de programación más fáciles de aprender, especialmente para aquellos que son nuevos en la programación. Su sintaxis simple y legible, junto con una curva de aprendizaje suave, permite a los principiantes adquirir rápidamente los conceptos fundamentales de la programación. Python también fomenta buenas prácticas de programación, como el uso de indentación para definir bloques de código, lo que promueve un código estructurado y legible.

Python ofrece numerosos beneficios en diversas áreas como desarrollo web, ciencia de datos, análisis de datos, aprendizaje automático, IA, creación de scripts y herramientas, procesamiento

de imágenes, entre otras. Su versatilidad, su amplia comunidad y su abundancia de bibliotecas especializadas hacen de Python una opción sólida para aquellos que buscan desarrollar habilidades en programación.

En el desarrollo web, Python cuenta con frameworks populares como Django y Flask, que facilitan la creación de aplicaciones web robustas y escalables. Su sintaxis clara y legible, así como su amplio soporte de bibliotecas, permiten desarrollar sitios web dinámicos y funcionales de manera eficiente.

En el campo de la ciencia de datos, Python se ha convertido en la opción preferida por su amplia gama de bibliotecas especializadas como NumPy, Pandas y Matplotlib, las cuales ofrecen herramientas poderosas para el procesamiento, el análisis y la visualización de datos, lo que facilita el descubrimiento de conocimientos valiosos y la toma de decisiones basadas en datos.

En el análisis de datos, Python proporciona capacidades robustas que permiten a los profesionales explorar y comprender conjuntos de datos complejos. Mediante el uso de bibliotecas como Pandas y herramientas de visualización como Matplotlib o Seaborn, los analistas de datos pueden hacer análisis estadísticos, crear gráficos informativos y extraer información significativa de los datos.

En el campo del aprendizaje automático (machine learning), Python es una opción popular debido a su amplio soporte de bibliotecas como Scikit-learn, TensorFlow y PyTorch, las cuales proporcionan algoritmos y herramientas para construir modelos de aprendizaje automático, realizar tareas de clasificación, de regresión, de clustering y de procesamiento de lenguaje natural, entre otros.

Por otra parte, Python es ampliamente utilizado en la implementación de algoritmos y técnicas de IA. Sus bibliotecas y frameworks, como Keras y TensorFlow, brindan una base sólida para desarrollar sistemas de IA sofisticados como reconocimiento de imágenes, procesamiento del lenguaje natural y sistemas de recomendación.

Python también es ideal para la creación de scripts y de herramientas de automatización. Su sintaxis sencilla y legible, combi-

nada con su amplio conjunto de bibliotecas estándar, permite a los desarrolladores crear scripts eficientes y herramientas personalizadas para automatizar tareas repetitivas y aumentar la productividad.

Python ofrece capacidades avanzadas de procesamiento de imágenes a través de bibliotecas como OpenCV y scikit-image que permiten manipular imágenes, aplicar filtros, realizar reconocimiento de objetos y hacer otras tareas de procesamiento de imágenes de manera eficiente.

En general, Python es un lenguaje de programación que ofrece muchas más características beneficiosas que lo convierten en una opción atractiva para una amplia gama de aplicaciones y proyectos de desarrollo de software.

Acerca del libro

Este libro es una guía para iniciar el proceso de aprendizaje de Python. En él, se presenta una introducción a los temas necesarios cuando se comienza a programar en Python. Abarca los fundamentos teóricos junto con ejemplos y código. En cada código se realiza una explicación del proceso que se lleva a cabo.

El objetivo es que los lectores comprendan los conceptos, escriban el código, lo ejecuten y lo modifiquen para adaptarlo a sus necesidades. Es importante que vayan desarrollando los ejercicios para poder comprender todos los componentes y darle sentido a la forma en que Python funciona.

Los fundamentos del lenguaje Python se presentan en pequeños pasos fáciles de entender. Después se profundiza en los objetos, las estructuras de datos y el manejo de archivos, que son conceptos básicos que se deben aprender al comenzar a programar.

Convenciones del libro

Las palabras clave de Python, las variables y los métodos desarrollados se resaltan en negrita y se ponen entre comillas dobles. El

código se presenta en el formato utilizado por Visual Studio Code. Después del código, se muestra el resultado generado durante el proceso de ejecución y se hace una explicación.

Capítulo 2

Instalación de Python