# Introducción



Todo programador tiene una historia sobre cómo aprendió a escribir su primer programa. Yo empecé a programar de 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 ordenador kit que mi padre había montado en nuestro sótano. El ordenador no era más que una placa base desnuda conectada a un teclado sin carcasa, y su monitor era un tubo de rayos catódicos desnudo. Mi programa inicial era un simple juego de adivinar números, que tenía este aspecto:

I'm thinking of a number! Try to guess the number I'm thinking of: 25  
Too low! Guess again: 50  
Too high! Guess again: 42  
That's it! Would you like to play again? (yes/no) no  
Thanks for playing!

Siempre recordaré lo satisfecho que me sentí, viendo a mi familia jugar a un juego que yo había creado y que funcionaba como yo pretendía.

Esa experiencia temprana 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 las que satisfacían mis esfuerzos infantiles, pero la sensación de satisfacción que me produce crear un programa que funciona sigue siendo en gran medida la misma.

## ¿A quién va dirigido este libro?

El objetivo de este libro es ponerte al día con Python lo más rápidamente posible para que puedas crear programas que funcionen -juegos, visualizaciones de datos y aplicaciones web-, al tiempo que desarrollas una base de programación que te servirá para el resto de tu vida. *Python Crash Course* está escrito para personas de cualquier edad que nunca hayan programado en Python o que nunca hayan programado en absoluto. Este libro es para quienes quieren aprender rápidamente los fundamentos de la programación para poder centrarse en proyectos interesantes, y para 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 dando una clase en la universidad y quieres una introducción a Python más amigable que el texto que te han asignado, este libro también puede facilitar tu clase. Si quieres cambiar de profesión, *Python Crash Course* puede ayudarte a hacer la transición a una carrera más satisfactoria. Ha funcionado bien para una gran 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. Aprenderás de forma eficaz y adoptarás buenos hábitos a medida que adquieras una base sólida en conceptos generales de programación. Después de abrirte camino a través de *Python Crash Course*, estarás preparado para pasar a técnicas más avanzadas de Python, y tu próximo lenguaje de programación será aún más fácil de comprender.

En la Parte I de este libro, aprenderás los conceptos básicos de programación que necesitas conocer para escribir programas en Python. Estos conceptos son los mismos que aprenderías al iniciarte en casi cualquier lenguaje de programación. Conocerás los distintos tipos de datos y las formas en que puedes almacenarlos en tus programas. Construirás colecciones de datos, como listas y diccionarios, y trabajarás con esas colecciones de forma eficiente. Aprenderás a utilizar los bucles while y las sentencias if para comprobar determinadas condiciones, de modo que puedas ejecutar secciones específicas de código mientras esas condiciones sean verdaderas y ejecutar otras secciones cuando no lo sean, una técnica que te ayuda a automatizar muchos procesos.

Aprenderás a aceptar entradas de los usuarios para que tus programas sean interactivos, y a mantener tus programas en ejecución tanto tiempo como desee el usuario. Explorarás cómo escribir funciones que hagan reutilizables partes de tu programa, de modo que sólo tengas que escribir bloques de código que realicen determinadas acciones una vez, mientras utilizas ese código tantas veces como necesites. Luego ampliarás este concepto a comportamientos más complicados con clases, haciendo que programas bastante sencillos respondan a diversas situaciones. Aprenderás a escribir programas que manejen errores comunes con elegancia. Después de trabajar cada uno de estos conceptos básicos, escribirás una serie de programas cada vez más complejos utilizando lo que has aprendido. Por último, darás tu primer paso hacia la programación intermedia aprendiendo a escribir pruebas para tu código, de modo que puedas seguir desarrollando tus programas sin preocuparte de introducir errores. Toda la información de la Parte I te preparará para emprender proyectos más grandes y complejos.

En la Parte II, aplicarás lo aprendido en la Parte I a tres proyectos. Puedes realizar cualquiera de estos proyectos o todos ellos, en el orden que mejor te convenga. En el primer proyecto, en los Capítulos 12-14, crearás un juego de disparos al estilo *Space Invaders*llamado *Alien Invasion*, que incluye varios niveles de juego cada vez más difíciles. Cuando hayas completado este proyecto, estarás bien encaminado para poder desarrollar tus propios juegos 2D. Incluso si no aspiras a convertirte en programador de juegos, trabajar en este proyecto es una forma agradable de unir gran parte de lo que aprenderás en la Parte I.

El segundo proyecto, en los Capítulos 15-17, te introduce en la visualización de datos. Los científicos de datos utilizan diversas técnicas de visualización para dar sentido a la enorme cantidad de información de que disponen. Trabajarás con conjuntos de datos que generes mediante código, conjuntos de datos que descargues de fuentes online y conjuntos de datos que tus programas descarguen automáticamente. Cuando hayas completado este proyecto, serás capaz de escribir programas que examinen grandes conjuntos de datos y hagan representaciones visuales de muchos tipos diferentes de información.

En el tercer proyecto, en los Capítulos 18-20, construirás una pequeña aplicación web llamada Registro de Aprendizaje. Este proyecto te permite llevar un diario organizado de la información que has aprendido sobre un tema concreto. Podrás mantener registros separados para diferentes temas y permitir que otras personas creen una cuenta e inicien sus propios diarios. También aprenderás a desplegar tu 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.](https://nostarch.com/python-crash-course-3rd-edition)

También mantengo un amplio conjunto de recursos complementarios en [https://ehmatthes.github.io/pcc\_3e.](https://ehmatthes.github.io/pcc_3e) Estos recursos incluyen lo siguiente:

Instrucciones deinstalación Las instrucciones de instalación en línea son idénticas a las del libro, pero incluyen enlaces activos en los que puedes hacer clic para seguir los distintos pasos. Si tienes algún problema de configuración, consulta este recurso.

Actualizaciones Python, como todos los lenguajes, evoluciona constantemente. Mantengo un minucioso conjunto de actualizaciones, así que si algo no funciona, comprueba aquí si han cambiado las instrucciones.

Soluciones a los ejercicios Deberías dedicar bastante tiempo a intentar por tu cuenta los ejercicios de las secciones "Pruébalo tú mismo". Sin embargo, si estás atascado y no consigues avanzar, las soluciones a la mayoría de los ejercicios están en línea.

Hojas detrucos También hay en línea un juego completo de hojas de trucos descargables para consultar rápidamente los conceptos principales.

## ¿Por qué Python?

Todos los años me planteo si seguir utilizando Python o pasarme a otro lenguaje, quizá uno más nuevo en el mundo de la programación. Pero sigo centrándome en Python por muchas razones. Python es un lenguaje increíblemente eficiente: tus 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". Tu código será más fácil de leer, más fácil de depurar y más fácil de ampliar y desarrollar, en comparación con otros lenguajes.

La gente utiliza Python con muchos fines: para hacer juegos, crear aplicaciones web, resolver problemas empresariales y desarrollar herramientas internas en todo tipo de empresas interesantes. Python también se utiliza mucho en el ámbito científico, para la investigación académica y el trabajo aplicado.

Una de las razones más importantes por las que sigo utilizando Python es 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 otros que ya han resuelto problemas similares. Tener una comunidad bien conectada y que te apoye es fundamental para ayudarte a resolver problemas, y la comunidad Python apoya plenamente a las personas que están aprendiendo Python como su primer lenguaje de programación o que llegan a Python con conocimientos previos de otros lenguajes.

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