

# Python para informáticos

Explorando la información

Version 2.7.2

Charles Severance

Copyright © 2009- Charles Severance.

Traducción al español por Fernando Tardío.

Historial de impresiones:

**Agosto 2015:** Primera edición en español de *Python para Informáticos: Explorando la información*.

**May 2015:** Permiso editorial gracia a Sue Blumenberg.

**Octubre 2013:** Revisión completa a los capítulos 13 y 14 para cambiar a JSON y usar OAuth. Añadido capítulo nuevo en Visualización.

**Septiembre 2013:** Libro publicado en Amazon CreateSpace

**Enero 2010:** Libro publicado usando la máquina Espresso Book de la Universidad de Michigan.

**Diciembre 2009:** Revisión completa de los capítulos 2-10 de *Think Python: How to Think Like a Computer Scientist* y escritura de los capítulos 1 y 11-15, para producir *Python for Informatics: Exploring the Information*

**Junio 2008:** Revisión completa, título cambiado por *Think Python: How to Think Like a Computer Scientist*.

**Agosto 2007:** Revisión completa, título cambiado por *How to Think Like a (Python) Programmer*.

**Abril 2002:** Primera edición de *How to Think Like a Computer Scientist*.

Este trabajo está licenciado bajo una licencia Creative Common Attribution-NonCommercial-ShareAlike 3.0 Unported. Esta licencia está disponible en [creativecommons.org/licenses/by-nc-sa/3.0/](http://creativecommons.org/licenses/by-nc-sa/3.0/). Puedes consultar qué es lo que el autor considera usos comerciales y no comerciales de este material, así como las exenciones de licencia en el apéndice titulado Detalles del Copyright.

Las fuentes L<sup>A</sup>T<sub>E</sub>X de la versión *Think Python: How to Think Like a Computer Scientist* de este libro están disponibles en <http://www.thinkpython.com>.

# Prefacio

## Python para informáticos: Remezclando un libro libre

Entre los académicos, siempre se ha dicho que se debe “publicar o morir”. Por ello, es bastante habitual que siempre quieran crear algo desde cero, para que sea su propia obra original. Este libro es un experimento que no empieza desde cero, sino que “remezcla” el libro titulado *Think Python: How to Think Like a Computer Scientist* (Piensa en Python: Cómo pensar como un informático), escrito por Allen B. Downey, Jeff Elkner, y otros.

En diciembre de 2009, yo estaba preparándome para enseñar **SI502 - Networked Programming** (Programación en red) en la Universidad de Michigan por quinto semestre consecutivo y decidí que ya era hora de escribir un libro de texto sobre Python que se centrara en el manejo de datos en vez de hacerlo en explicar algoritmos y abstracciones. Mi objetivo en SI502 es enseñar a la gente habilidades para el manejo cotidiano de datos usando Python. Pocos de mis estudiantes planean dedicarse de forma profesional a la programación informática. La mayoría esperan llegar a ser bibliotecarios, administradores, abogados, biólogos, economistas, etc., aunque quieren aplicar con eficacia la tecnología en sus respectivos campos.

Como no conseguía encontrar un libro orientado a datos en Python adecuado para mi curso, me propuse escribirlo yo mismo. Por suerte, en una reunión de la facultad tres semanas antes de que empezara con el nuevo libro (que tenía planeado escribir desde cero durante las vacaciones), el Dr. Atul Prakash me mostró el libro *Think Python* (Piensa en Python) que él había usado para su curso de Python ese semestre. Se trata de un texto sobre ciencias de la computación bien escrito, con explicaciones breves y directas y fácil de entender.

La estructura general del libro se ha cambiado para conseguir llegar a los problemas de análisis de datos lo antes posible, y contiene, casi desde el principio, una serie de ejemplos y ejercicios con código, dedicados al análisis de datos.

Los capítulos 2–10 son similares a los del libro *Think Python*, pero en ellos hay cambios importantes. Los ejemplos y ejercicios dedicados a números han sido reemplazados por otros orientados a datos. Los temas se presentan en el orden adecuado para ir construyendo soluciones de análisis de datos progresivamente más sofisticadas. Algunos temas, como `try` y `except`, se han adelantado y son

presentados como parte del capítulo de condicionales. Las funciones se tratan muy someramente hasta que se hacen necesarias para manejar programas complejos, en vez de introducirlas en las primeras lecciones como abstracción. Casi todas las funciones definidas por el usuario han sido eliminadas del código de los ejemplos y ejercicios, excepto en el capítulo 4. La palabra “recursión”<sup>1</sup> no aparece en todo el libro.

En los capítulos 1 y 11–16, todo el material es nuevo, centrado en el uso con problemas del mundo real y en ejemplos sencillos en Python para el análisis de datos, incluyendo expresiones regulares de búsqueda y análisis, automatización de tareas en el PC, recepción de datos a través de la red, rastreo de páginas web en busca de datos, uso de servicios web, análisis de datos XML y JSON, y creación y uso de bases de datos mediante el lenguaje de consultas estructurado (SQL).

El objetivo final de todos estos cambios es pasar de un enfoque de ciencias de la computación a uno puramente informático, incluyendo solamente temas de tecnología básica que puedan ser útiles incluso si los alumnos al final eligen no convertirse en programadores profesionales.

Los estudiantes que encuentren este libro interesante y quieran adentrarse más en el tema deberían echar un vistazo al libro de Allen B. Downey *Think Python*. Gracias a que hay muchos temas comunes en ambos libros, los estudiantes adquirirán rápidamente habilidades en las áreas adicionales de la programación técnica y razonamiento algorítmico que se tratan en *Think Python*. Y dado que ambos libros tienen un estilo similar de escritura, deberían ser capaces de moverse rápidamente por *Think Python* con un mínimo de esfuerzo.

Como propietario de los derechos de *Think Python*, Allen me ha dado permiso para cambiar la licencia del material de su libro que aparece también en éste, desde la GNU Free Documentation License (Licencia de Documentación Libre) a la más reciente Creative Commons Attribution – Share Alike license. Esto sigue un cambio general en las licencias de documentación abierta, que están pasando del GFDL al CC-BY-SA (como, por ejemplo, Wikipedia). El uso de la licencia CC-BY-SA mantiene la tradicional fortaleza del copyleft a la vez que hace que sea más sencillo para los autores nuevos el reutilizar este material como les resulte más provechoso.

Creo que este libro sirve como ejemplo de por qué los materiales libres son tan importantes para el futuro de la educación, y quiero agradecer a Allen B. Downey y al servicio de publicaciones de la Universidad de Cambridge por su amplitud de miras al permitir que este libro esté disponible con unos derechos de reproducción abiertos. Espero que estén satisfechos con el resultado de mis esfuerzos y deseo que tú como lector también estés satisfecho con *nuestros* esfuerzos colectivos.

Quiero agradecer a Allen B. Downey y a Lauren Cowles su ayuda, paciencia y orientación en la gestión y resolución del tema de los derechos de autor en torno a este libro.

---

<sup>1</sup>Excepto, por supuesto, en esta línea.

Charles Severance  
www.dr-chuck.com  
Ann Arbor, MI, USA  
9 de Septiembre de 2013

Charles Severance es un profesor clínico asociado en la *School of Information* de la Universidad de Michigan.



# Índice general

<b>Prefacio</b>	<b>III</b>
chapter1. ¿Por qué debería aprender a escribir programas?1	
1.1. Creatividad y motivación . . . . .	2
section1.2.Arquitectura hardware del PC3	
1.3. Comprendiendo la programación . . . . .	5
section1.4.Palabras y frases5	
1.5. Conversando con Python . . . . .	6
section1.6.Terminología: intérprete y compilador9	
1.7. Escribir un programa . . . . .	11
section1.8.¿Qué es un programa?11	
1.9. Los bloques de construcción de los programas . . . . .	13
section1.10.¿Qué es posible que vaya mal?14	
1.11. El viaje de aprendizaje . . . . .	15
section1.12.Glosario16	
1.13. Ejercicios . . . . .	17
chapter2. Variables, expresiones y sentencias21	
2.1. Valores y tipos . . . . .	21
section2.2.Variables22	
2.3. Nombres de variables y palabras claves . . . . .	23
section2.4.Sentencias24	
2.5. Operadores y operandos . . . . .	24
section2.6.Expresiones25	

2.7. Orden de las operaciones . . . . .	25
section2.8.Operador módulo	26
2.9. Operaciones con cadenas . . . . .	27
section2.10Petición de información al usuario	27
2.11. Comentarios . . . . .	28
section2.11Elección de nombres de variables mnemónicos	29
2.13. Depuración . . . . .	31
section2.14Glosario	31
2.15. Ejercicios . . . . .	32
chapter3. Ejecución condicional	35
3.1. Expresiones booleanas . . . . .	35
section3.2.Operadores lógicos	36
3.3. Ejecución condicional . . . . .	36
section3.4.Ejecución alternativa	37
3.5. Condicionales encadenados . . . . .	38
section3.6.Condicionales anidados	39
3.7. Captura de excepciones usando try y except . . . . .	40
section3.8.Evaluación en cortocircuito de expresiones lógicas	42
3.9. Depuración . . . . .	43
section3.10Glosario	44
3.11. Ejercicios . . . . .	45
chapter4. Funciones	47
4.1. Llamadas a funciones . . . . .	47
section4.2.Funciones internas	47
4.3. Funciones de conversión de tipos . . . . .	48
section4.4.Números aleatorios	49
4.5. Funciones matemáticas . . . . .	50
section4.6.Añadiendo funciones nuevas	51
4.7. Definición y usos . . . . .	52