

El pequeño libro de algoritmos

2019-08-29

El pequeño libro de algoritmos

Este libro escrito por William Lau está diseñado para ayudar a maestros y estudiantes a desarrollar fluidez en programación en el lenguaje Python. La inspiración para escribir este libro nace de la pregunta ¿Por qué es tan difícil aprender a programar?. Está dirigido a quienes ya han desarrollado habilidades básicas en las tres estructuras fundamentales de la programación: secuencia, decisión e iteración.

Este libro escrito por William Lau está diseñado para ayudar a maestros y estudiantes a desarrollar fluidez en programación en el lenguaje Python. La inspiración para escribir este libro nace de la pregunta ¿Por qué es tan difícil aprender a programar?. Está dirigido a quienes ya han desarrollado habilidades básicas en las tres estructuras fundamentales de la programación: secuencia, decisión e iteración.

EL PEQUEÑO LIBRO DE ALGORITMOS

<http://eduteka.icesi.edu.co/pdfdir/lau-the-little-book-of-algorithms.pdf> Este libro está diseñado para ayudar a maestros y estudiantes a desarrollar fluidez en programación en el lenguaje [Python](#). Está dirigido a quienes ya han desarrollado habilidades básicas en las tres estructuras fundamentales de la programación: secuencia, decisión e iteración. El objetivo original del autor era ayudar a sus estudiantes de grado 11° con el examen “[GCSE Computer Science](#)”. Sin embargo, el libro resulta útil para estudiantes y educadores que no estén interesados en dicha examinación. La razón para seleccionar Python fue lo popular del lenguaje y la facilidad que representa para

los estudiantes nuevos en este entorno de programación.

La inspiración para escribir este libro nace de la pregunta "**¿Por qué es tan difícil aprender a programar?**". El autor recordó como para él mismo fue desafiante la programación en sus inicios y como se establece una ruta de aprendizaje en esta disciplina que no termina nunca. Después de enseñar programación durante los últimos siete años, William Lau notó que solo una minoría de sus estudiantes se sentían lo suficientemente seguros como para programar independientemente después de dos años de clases; entonces, cambió su forma de enseñar.

Él cree que [Scott Portnoff](#) está en lo correcto: los estudiantes necesitan memorizar algunas estructuras clave de programación: sentencias "Sí... entonces", bucles tipo "mientras" y bucles tipo "para", por citar algunas. Esto disminuye la carga cognitiva y permite a los estudiantes programar con mayor fluidez. El trabajo de Portnoff fue su punto de partida para este libro. De estudios culturales tomó la idea de canon y pensó en crear un canon para programadores a partir de la pregunta: **¿Quizás hay un conjunto de programas que representan algoritmos con los que cada estudiante de informática debería familiarizarse?**

Comencé a compilar una lista de programas cortos basados en mi experiencia tanto de profesor como de evaluador, con el fin que los estudiantes los memorizaran. Aunque aprender de memoria programas largos es desafiante e inútil, los más cortos pueden ayudar a los estudiantes a desarrollar fluidez y a comprender esos programas al aplicarlos.

Sue Sentance sugirió en una introducción a los cursos de programación que se debería presentar a los estudiantes las subrutinas desde su primer programa. Por su parte, Richard Pawson va un paso más allá en la [edición 07 de la revista Hello World](#); en esa edición Pawson presenta un caso de programación usando el paradigma de programación de funciones desde la primera clase. Él presenta argumentos sólidos para usar funciones que devuelven valores en lugar de contener entradas y salidas. Esto parece contrario a la intuición debido a la complejidad percibida de la sintaxis de las funciones; sin embargo, existen tres argumentos clave para usar funciones: a) la prueba de unidad en las funciones individuales, b) reutilización de código y c) división de tareas. Por lo tanto, Lau recomienda escribir programas con funciones desde el

principio. Esto parece desalentador al comienzo; sin embargo, la repetición conducirá a la fluidez.

A pesar de las ventajas irrefutables de programar con funciones, el autor incluye también procedimientos (subrutinas que no devuelven valores) y programas que no usan subrutinas en absoluto.

En relación al uso del libro, Lau recomienda en cada clase pasar en detalle por uno o dos algoritmos, hablando de los programas línea por línea. Luego pedirles a los estudiantes que vayan al desafío asociado al final del libro. Una vez que han redactado el programa a mano, pueden verificar si el programa se ejecuta correctamente con un entorno de programación. Alternativamente, los estudiantes pueden optar por trabajar el libro independientemente en clase o en casa. Como todos los libros de ejercicios, las respuestas se proporcionan en la parte final.

CRÉDITOS

Traducción al español realizada por Eduteka de algunos apartes del libro "[The Little Book of Algorithms](#)", escrito por William Lau (@MrLauLearning) y publicado bajo la licencia Creative Commons Attribution - NonCommercial - ShareAlike 4.0 International License. William lau (william.lau@computingschool.org.uk) lidera la Etapa 4 de Computación en la "[Central Foundation Boys' School](#)". Además es autor de [Teaching Computing in Secondary Schools](#) y [The Little Book of Algorithms](#).

Publicación de este documento en EDUTEKA: Agosto 29 de 2019.

Última actualización de este documento: Agosto 29 de 2019.