PYTHON APLICADO

Book · March 2021

CITATIONS
0

READS
3,694

1 author:

Eugenia Bahit
Bahit & Bahit Ltd
25 PUBLICATIONS **1** CITATION

SEE PROFILE

PYTHON APLICADO

EUGENIA BAHIT

Informática Teórica[†]
Elida Bahit Research Centre

EBRC Publisher

Londres • Madrid • Buenos Aires • México• Bogotá
2021

[†] Directora del Elida Bahit Research Centre de Reino Unido. Contacto: www.elidabahit.co.uk

PYTHON APLICADO

ISBN: 978-1-8381901-2-5
e-ISBN: 978-1-8381901-3-2

Título original: Python para principiantes, Edición 2020
Publicado en Reino Unido por EBRC Publisher
71-75 Shelton Street, Covent Garden, London, WC2H 9JQ
www.elidabahit.co.uk

Editora: Eugenia Bahit

Diseño interior: Amanda Kraft
Diseño de portada: Julia Davies

Impreso en:

Argentina por Docuprint S.A.
para distribución exclusiva en el territorio argentino, Chile, Paraguay y Uruguay.

Colombia, por Bubok Publishing S.L.
para distribución exclusiva en territorio colombiano.

España, por Bubok Publishing S.L.
para distribución exclusiva en territorio español.

México, por Bubok Publishing S.L.
para distribución exclusiva en territorio mexicano.

Estados Unidos, por Lulu Press, Inc.
para distribución en América Latina (excepto Argentina), Reino Unido, Unión Europea, Asia, África y Oceanía.

A quienes construyen sus sueños con pasión y se esfuerzan para alcanzar sus metas.

Índice de contenidos

Prólogo: entender la diferencia entre programar y codificar

En el ámbito de la ingeniería de software, un *problema* es un requerimiento concreto que necesita ser resuelto mediante un programa informático, mientras que su *análisis* consiste en entender qué se necesita hacer para determinar cómo hacerlo, mediante qué operaciones, con cuáles valores de entrada y para arrojar cuáles valores de salida. Los *algoritmos* constituyen el paso a paso de la solución del problema.

Como características principales de los algoritmos, puede decirse qué:

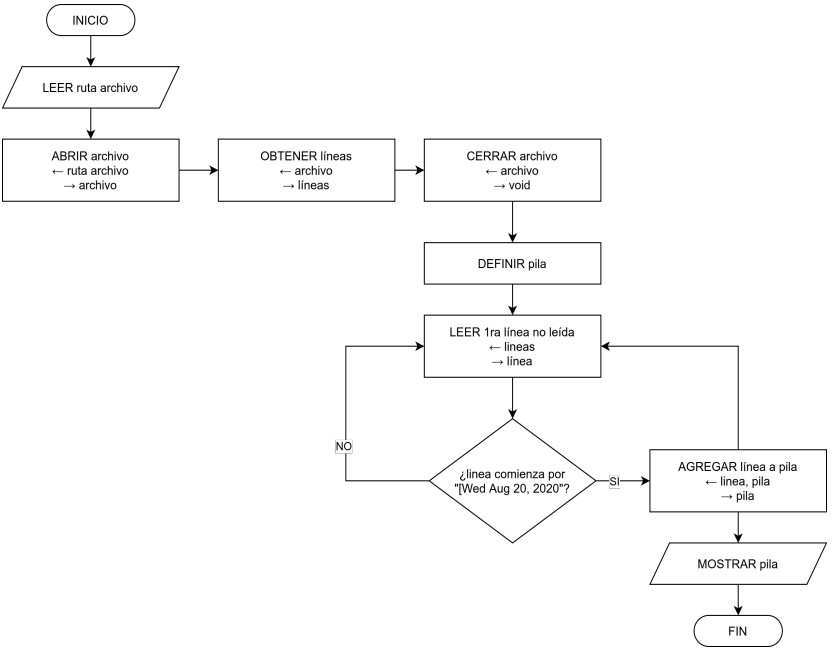
1. comienzan aceptando los datos de entrada que serán procesados;
2. las instrucciones definidas deben ser completas, precisas y concretas;
3. el tiempo total empleado por un algoritmo para llevar a cabo sus operaciones debe ser finito, así como la cantidad de instrucciones y el número de repeticiones cuando las hubiera;
4. finalmente, debe producir al menos un resultado (valor de salida).

Un factor clave de la programación es diseñar los algoritmos con una completa abstracción del lenguaje. Por ejemplo, si tras el análisis de un problema se determina que es necesario visualizar una pila con las entradas del registro de errores del archivo *errors.log* del servidor HTTP de Apache del día jueves 20 de agosto de 2020, considerando que cada

entrada del registro necesaria es una línea que comienza por `[Thu Aug 20, 2020`, se podría resolver, independientemente del lenguaje a utilizar, con un algoritmo como el siguiente:

```
1. LEER ruta del archivo
2. ABRIR archivo
3. OBTENER líneas del archivo leído
4. CERRAR archivo
5. DEFINIR pila
6. SI la primera línea no leída comienza por "[Wed Aug 20,
   2020"
   AGREGAR línea a pila
7. Repetir el paso 6 por cada línea no leída hasta que no
   queden líneas sin leer
8. MOSTRAR la pila en pantalla
```

Para facilitar la comprensión de un algoritmo, puede representarse en un *diagrama de flujo* como el siguiente:



Estos diagramas emplean diversas figuras tales como:

- *paralelogramos*, para los valores de entrada y salida;
- *rectángulos*, para las operaciones;
- *rombos*, para los condicionales;
- y *rectángulos con bordes redondeados*, para indicar inicio y fin del algoritmo.

Posteriormente, los algoritmos son traducidos a código fuente, y aquellas operaciones no disponibles de forma nativa en el lenguaje, requerirán el diseño de un nuevo algoritmo.

El siguiente código es la traducción correspondiente al diagrama de flujo anterior:

```
def obtener_registros(ruta_archivo):
    archivo = open(ruta_archivo, "r")
    lineas = archivo.readlines()
    archivo.close()
    pila = []
    for linea in lineas:
        if linea.startswith("[Wed Aug 20, 2020"):
            pila.append(linea)

    print(pila)
```

El *código fuente* es el conjunto de órdenes escritas en un determinado lenguaje de programación, que componen un programa. Los *lenguajes de programación* son lenguajes informáticos diseñados para que un ordenador pueda llevar a cabo operaciones concretas. Los *lenguajes informáticos* son lenguajes artificiales creados exclusivamente para ser entendidos por los ordenadores. Los de programación son los que permiten la ejecución de órdenes pero los hay de otros tipos. Se manejan en dos niveles básicos: *bajo nivel*, que se refiere al código

máquina (código binario) y los de *alto nivel* que poseen una estructura humanamente legible e independiente del hardware.

Los hay *interpretados* (sus instrucciones son ejecutadas directamente sobre el sistema operativo por un intérprete) y *compilados* (aquellos que requieren ser traducidos a un lenguaje intermedio y empaquetados para su ejecución). Algunos funcionan en diversas plataformas y otros son exclusivos de sistemas concretos. Algunos aceptan solo un paradigma (técnica) de programación, mientras que otros, pueden aceptar más de uno.

Si bien al momento de traducir un algoritmo a código fuente la sintaxis empleada dependerá exclusivamente del lenguaje de programación elegido, todo lenguaje posee una serie de elementos comunes —a todos los lenguajes— que se encuentran disponibles. Solo la sintaxis del lenguaje es lo que cambia. Por ejemplo, mientras que una variable *foo* de tipo entero con valor 15, en Python se define mediante `foo = 15`, en PHP se define anteponiendo un signo dólar al nombre de la variable y un punto y coma tras el valor, tal que `$foo = 15;` y en C, requiere la declaración de tipo, tal que `int foo = 15;`.

El proceso de traducción de un algoritmo a código fuente en un lenguaje de programación determinado es lo que técnicamente se conoce como ***codificación***. El proceso que abarca el análisis de un problema, el diseño de su solución (algoritmo) más su traducción a código fuente (codificación), es lo que se denomina ***programación***. Así, el estudio de esta abarca tres grandes temas: lógica, análisis y diseño de algoritmos; y lenguajes de programación. ***«Python Aplicado»*** pertenece a esta tercera categoría.