

Introducción a Python

Módulo 1

¿Qué es la programación?

¿Qué es la programación?

Entendemos por **programación** la actividad de programar; y por *programar*, el decirle a una computadora — o, mejor dicho, cualquier dispositivo programable — lo que tiene que hacer.

De modo que la programación es un tipo de **comunicación**. Comunicación entre un humano (programador) y una máquina (la que es programada). Y como todo tipo de comunicación, precisa de un lenguaje que sea comprendido tanto por el emisor como por el receptor.

A ese lenguaje lo denominamos ***lenguaje de programación***. El mensaje codificado según las reglas de un lenguaje de programación lo llamamos ***programa o aplicación***.



Existen muchos lenguajes de programación, esto es, como dijimos, medios de comunicación entre un programador y una computadora.

Cada uno de ellos se caracteriza por algo en particular en el proceso de “instruir” a una máquina para que ejecute las acciones que le indicamos. Algunos lenguajes son más óptimos o acordes que otros para tal o cual tarea.

Así, hay lenguajes que están pensados para “hablarle” a un celular; otros, a una computadora de escritorio; otros, a un microondas.

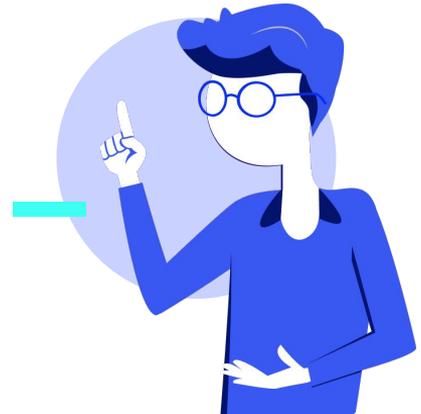
Y algunos son más fáciles de aprender, del mismo modo que decimos que el griego y el árabe son menos accesibles que el inglés.

El inglés, no obstante, es más acorde si nos queremos dedicar al comercio internacional; así como el griego lo es para el estudio de los pensadores clásicos de la antigüedad.



Seamos un poco más precisos: no es cierto que las computadoras comprenden las instrucciones que les comunicamos vía un lenguaje de programación. De hecho, están diseñadas para entender un dialecto que se denomina *código máquina*.

Escribir un conjunto de instrucciones directamente en este código es simplemente imposible para un ser humano. Por ende, necesitaremos un “traductor” (que no es sino un programa) que convierta nuestras instrucciones escritas según las reglas de un lenguaje de programación en particular a código máquina.



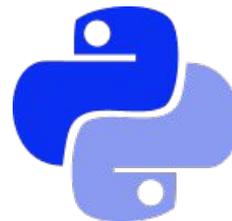
Python

***Python* es un lenguaje de programación.**

Pero también llamamos *Python* al programa que traduce Python -el lenguaje- a código máquina para que pueda ser interpretado por una computadora.

De ahí que en la expresión “instalar Python” nos referimos a Python -el programa-, mientras que “programar en Python” se refiere a Python -el lenguaje-.

En la jerga, para evitar dicha ambigüedad, a Python -el programa- también se lo denomina *intérprete*.



Python es un lenguaje que se caracteriza por tener una **sintaxis sencilla** (fácil de aprender para legos en programación) y clara (elocuente y de lectura fluida). Por ello es una herramienta ideal para iniciarse en la programación.

Independientemente del lenguaje que usaremos en este curso, lo más importante es que vamos a aprender los **conceptos generales** de la programación (aunque claro, con las particularidades de Python), que se aplicarán igualmente para otros.



Así como aprender un segundo idioma es más sencillo habiendo dominado nuestra lengua materna, lo mismo ocurre en el mundo de la programación. No es de extrañar que una persona, con el tiempo, sea versada en varios lenguajes de programación; aunque por lo general uno termina tomando una decisión según sus propios gustos y especializándose en alguno en particular.

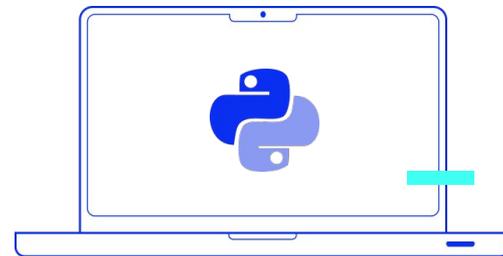


Para finalizar este apartado, precisemos una clasificación que nos será útil para avanzar.

Ya dijimos que un **programa** o **aplicación** es un **conjunto de instrucciones escritas** según las reglas de un lenguaje de programación. Si bien existen varios tipos de aplicaciones, en este curso estaremos trabajando con los siguientes dos:

- Aplicaciones de consola.
- Aplicaciones de escritorio.

Por el momento basta con saber que en Python los programas son, por defecto, aplicaciones de consola.



Estructura de un programa

Estructura de un programa

Nos permitimos usar la siguiente analogía (bastante trillada): *escribir un programa, es decir programar, es como redactar una receta de cocina*. La diferencia es que quien escribe es un programador y quien “cocina”, una máquina.

Las computadoras, a diferencia de los cocineros, nunca se equivocan al interpretar una “receta”. Hacen lo que les decimos, al pie de la letra.

Todos los problemas que surgen en la industria del software radican en creer que le estamos indicando una actividad “X” a una máquina cuando, en realidad, programamos una actividad “Y”. Esto es, cuando hay una disociación entre lo que le “decimos” y lo que quisimos decir.

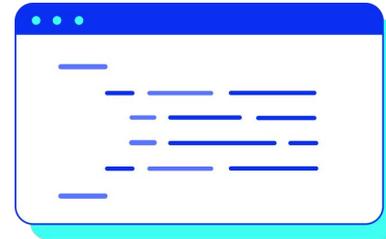


De esta definición deducimos que podemos escribir un programa tanto en un pizarrón como en un programa pensado para esa tarea -al que llamamos *editor de código*-.

Las instrucciones de los programas tienen una disposición similar a la del español: se interpretan de arriba a abajo y de izquierda a derecha.

Aunque a veces podemos indicarle que queremos que una instrucción o conjunto de instrucciones se ejecuten dos o más veces, o que en ocasiones no se ejecute, o que se vuelva a la primera instrucción, entre otras cosas similares.

Este recorrido del código (entendemos código por conjuntos de instrucciones) que hace una máquina al interpretarlo se denomina *flujo de un programa*.



Un conjunto de instrucciones dado con el objetivo de obtener un determinado comportamiento se conoce como **algoritmo** (según el diccionario de la RAE conjunto ordenado y finito de operaciones que permite hallar la solución de un problema).

Por ejemplo, el proceso por el cual se calculan las raíces de una ecuación de segundo grado constituye un algoritmo.

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Si tuviésemos que describirlo, sería algo así:

1. Elevar **b** al cuadrado.
2. Multiplicar 4 por **a** y por **c**.
3. Restar el resultado de (2) al resultado de (1).
4. Calcular la raíz cuadrada del resultado de (3).
5. Sumar el resultado de (4) **a - b**.
6. Dividir el resultado de (5) por **2a**.
7. Restar el resultado de (4) **a - b**.
8. Dividir el resultado de (7) por **2a**.

Si bien hemos omitido algunos pasos (como por ejemplo, qué ocurre si el resultado del paso 3 es negativo), la idea es ilustrar las operaciones que conforman un algoritmo y el orden en el que están dispuestas.

**¡Sigamos
trabajando!**