Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.

Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License.

Python es un lenguaje de programación multiparadigma. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional. Otros paradigmas están soportados mediante el uso de extensiones.

Python usa tipado dinámico y conteo de referencias para la administración de memoria.

Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos).

Otro objetivo del diseño del lenguaje es la facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en C o C++.

Conceptos previos

Programa: Conjunto de instrucciones que entiende una computadora para realizar una actividad.

Todo programa tiene un objetivo bien definido: un procesador de texto es un programa que permite cargar, modificar e imprimir textos, un programa de ajedrez permite jugar al ajedrez contra el ordenador u otro contrincante humano.

La actividad fundamental del programador es resolver problemas empleando el ordenador como herramienta fundamental.

Variable: Es un depósito donde hay un valor. Consta de un nombre y pertenece a un tipo de dato.

Tipos de variables

- Valores Enteros (100, 260, etc.)

- Valores Reales (1.24, 2.90, 5.00, etc.)
- Cadenas de caracteres ("Ejemplo", "De ", "Prueba", etc.)

Condicionales :

Los condicionales if-else, son una estructura de control, que nos permiten tomar cierta decisión al interior de nuestro algoritmo, es decir, nos permiten determinar que acciones tomar dada o no cierta condición, por ejemplo determinar si la contraseña ingresada por el usuario es válida o no y de acuerdo a esto darle acceso al sistema o mostrar un mensaje de error.

En resumen, un condicional if-else es una estructura que nos posibilita definir las acciones que se deben llevar a cabo si se cumple cierta condición y también determinar las acciones que se deben ejecutar en caso de que no se cumpla.

Ciclo For :

Los ciclos for son lo que se conoce como estructuras de control de flujo cíclicas o simplemente estructuras cíclicas, estos ciclos, como su nombre lo sugiere, nos permiten ejecutar una o varias líneas de código de forma iterativa, conociendo un valor específico inicial y otro valor final, además nos permiten determinar el tamaño del paso entre cada "giro" o iteración del ciclo.

En resumen, un ciclo for es una estructura de control iterativa, que nos permite ejecutar de manera repetitiva un bloque de instrucciones, conociendo previamente un valor de inicio, un tamaño de paso y un valor final para el ciclo.

Ciclo While

Los ciclos while son también una estructura cíclica, que nos permite ejecutar una o varias líneas de código de manera repetitiva sin necesidad de tener un valor inicial e incluso a veces sin siquiera conocer cuando se va a dar el valor final que esperamos, los ciclos while, no dependen directamente de valores numéricos, sino

de valores booleanos, es decir su ejecución depende del valor de verdad de una condición dada, verdadera o falso, nada más. De este modo los ciclos while, son mucho más efectivos para condiciones indeterminadas, que no conocemos cuándo se van a dar a diferencia de los ciclos for, con los cuales se debe tener claro un principio, un final y un tamaño de paso.

Funciones

Las funciones son una herramienta indispensable para el programador, tanto las funciones creadas por él mismo como las que le son proporcionadas por otras librerías, cualquiera que sea el caso, las funciones permiten automatizar tareas repetitivas, encapsular el código que utilizamos, e incluso mejorar la seguridad, confiabilidad y estabilidad de nuestros programas. Dominar el uso de funciones es de gran importancia, permiten modularizar nuestro código, separarlo según las tareas que requerimos, por ejemplo una función para abrir, otra para cerrar, otra para actualizar, etc. básicamente una función en nuestro código debe contener la implementación de una utilidad de nuestra aplicación, es decir que por cada utilidad básica (abrir, cerrar, cargar, mover, etc.) sería adecuado tener al menos una función asociada a ésta.