

Bitácora 4 del Taller de Herramientas Computacionales

Elías Jiménez Cruz, 409085596

12/01/2019

En la cuarta clase, comenzamos a ver el entorno de Python. La versión de Python que debemos usar es Python 2, en mi caso, utilizaré la versión 2.7.13. Muchas distribuciones de Linux tienen preinstalada alguna versión de Python, para comprobarlo basta con ir a Bash y escribir "python" como comando y presionar dos veces la tecla tabulador. Ahí se mostrarán todas las versiones de Python ya instaladas en el sistema. Ahora bien, lo que está instalado es el lenguaje únicamente, no se encuentra instalado algún editor adecuado para hacer scripts de programas escritos en ese lenguaje. Si bien es posible escribir un programa de Python en el editor de texto preinstalado en el sistema, es recomendable usar un editor adecuado, cuya interfaz sea amigable para escribir un programa, es decir, que tenga elementos que permitan detectar errores, completar paréntesis o comandos, resaltar con colores determinados comandos y palabras reservadas para distinguirlos fácilmente. En el caso de la clase, se utilizará el editor Idle, que es un IDE o entorno de desarrollo integrado, que a fin de cuentas es una aplicación que proporciona servicios para facilitar el desarrollo de software. Para instalar Idle, se utilizó el comando "apt install idle", el cual se encargó de todo el proceso. Ahora bien, es posible correr Python desde Bash, basta con ejecutar el comando "python", pero con este método se accede al intérprete de una forma muy básica, a nivel consola, no muy apto para escribir y guardar scripts. En vez de ello, simplemente se va a ejecutar el comando idle, el cual abrirá una ventana con la consola de Python, pero desde la que se podrán

abrir archivos nuevos y existentes, editarlos, guardarlos y ejecutarlos cómodamente desde una interfaz gráfica. Así, una vez conocido nuestro nuevo entorno de trabajo, pasamos a averiguar cómo resolver un problema desde la perspectiva vista de la clase pasada. El problema en cuestión fue cómo calcular la posición de una pelota en un momento determinado, cuando es lanzada a una velocidad inicial hacia arriba. Esto es claramente un problema que se resuelve con la fórmula de la caída libre, donde la posición deseada se expresa como:

$$Posicion = (VelocidadInicial)(tiempo) - \frac{(gravedad)(tiempo^2)}{2} \quad (1)$$

¿Pero cómo se haría un programa en Python que calcule, por ejemplo, la posición del objeto cuando la velocidad inicial es $34 \frac{m}{s}$, en el segundo cinco después de que fuera lanzado y considerando a la aceleración de la gravedad como $9.81 \frac{m}{s^2}$? Para ello tuvimos que aprender el uso en Python de las operaciones básicas, que son multiplicación, suma, división, resta y potencia, en vistas de poder escribir la fórmula deseada en un programa. A continuación, en Idle creamos el programa Ejemplo1Pelota.py mediante la herramienta Nuevo archivo del menú Archivo y lo guardamos en la carpeta de Programas, con lo cual escribimos el programa declarando el valor de las variables que representarían a la velocidad inicial, a la gravedad y al tiempo, y las combinamos en la fórmula deseada, cuyo valor lo asignamos a la variable posición, que sería la que sería imprimida por el programa mediante el comando print. Así, se guardó el programa y se presionó la tecla f5 para que fuera ejecutado en idle, el cual arrojó un resultado incorrecto, producto de un pequeño detalle que hay que tener en cuenta a la hora de programar en esta versión del lenguaje: el programa arroja los resultados conforme al tipo de variable que se le da. Es decir, si en una operación se le ordena sumar dos números enteros, el programa arrojará como resultado un número entero. Esto representa un problema a la hora de dividir, pues si se le ordena dividir dos números enteros, el programa arrojará un número entero, sin importar si el resultado no lo era. Para corregir este problema, basta con ingresar alguno de los números que componen la división como

un número de tipo flotante, es decir, como un número que tenga decimales, aunque estos sean cero. Python distingue a 2 como número entero y a 2.0 como número flotante, simplemente es el tipo de variable en el que se guarda la información. De esta manera, se corrigió el error en el programa y en esta ocasión arrojó el resultado deseado al ejecutarlo. Resultó un gran avance en el curso, al grado de que hasta se dejó como tarea hacer un programa que resolviera el cálculo de alguna otra fórmula matemática. En mi caso, escogí el teorema de Pitágoras.