



Unidad Azcapotzalco

División de Ciencias y Artes para el Diseño

Posgrado en Diseño y Visualización de la Información

Empleo de Python

L. C. C. Eric Omar Torres Velasco

Ensayo de investigación para la UEA Introducción a la programación

Profesora:

Dra. Lizbeth Gallardo López

Ciudad de México, México. A 23 de mayo de 2019

Python es un lenguaje de programación de código abierto lanzado en el año de 1991. Nace de la mano del programador holandés Guido van Rossum, con el propósito de diseñar un lenguaje más conciso en las expresiones pero sin las propiedades interpretadas en términos de legibilidad.

Recientemente, existen diversas tareas computacionales las cuales python puede realizar, vr. g., crear aplicaciones de lado del servidor, crear flujos de trabajo, conectarse a sistemas de bases de datos, manejar big data y operar matemáticas complejas, prototipar de manera rápida y ágil o producir la fase de desarrollo de un software, entre otras. Además, funciona en diferentes plataformas operativas (windows, mac, linux, raspberry pi, etc.), así como de tratarse de forma procesal, orientada a objetos o funcional.

Para instalar el intérprete de python en el sistema operativo de mac se presenta un breve manual con las siguientes instrucciones:

1. Entrar al siguiente enlace <https://www.python.org/>
2. Dar clic en “descargas” y seleccionar “Mac OS X”
3. Dar clic en “[Latest Python 3 Release - Python 3.7.3](#)”
4. Después seleccionar “[macOS 64-bit installer](#)” para descargar el archivo “python-3.7.3-macosx10.9.pkg”
5. Al terminar la descarga, dar doble clic sobre el archivo para abrirlo.
6. Seguir las instrucciones del instalador wizard con el botón continuar, aceptar las condiciones del contrato de licencia de software e instalar el software en el disco de almacenamiento principal.

A continuación, se abrirá la ubicación con el directorio “Python 3.7” listo para llamar al intérprete de python o ejecutar un programa desde la línea de comandos. Para esto, primero se necesita ubicar la ruta del directorio donde se encuentra el archivo fuente (.py) por medio del comando “cd”, después se llama al intérprete escribiendo “python” seguido del nombre del archivo fuente (p. ej., python filename.py). Es importante que exista un espacio de la barra espaciadora entre el comando y el nombre del archivo con la extensión de python. Por otra parte, para llamar al

intérprete de python, basta con escribir “python” en la línea de comando de mac (terminal) y comenzar a escribir la sintaxis propia del lenguaje y dar “enter” para verlo funcionar.

El editor de textos que emplearé para construir programas en python será Sublime.

La indentación en python es muy importante, debido a que, a diferencia de otros lenguajes, no utiliza puntos y comas para separar las líneas de instrucciones ni tampoco llaves o paréntesis para la declaración de estructuras de control. Como lo muestra el siguiente ejemplo:

```
if True:
    print("hola mundo!")
```

En python existen maneras de colocar un comentario. La primera opción es utilizando el signo de “#” seguido del texto, la segunda opción se utiliza más para comentarios multilínea y documentación, ésta indica que el texto debe estar entre triple doble comillas (p. ej., “""" texto """”).

El uso de variables como en cualquier lenguaje de programación, almacenan valores de tipo número, enteros, flotantes, double, complejos, booleano, carácter, cadena, etc. Las recomendaciones para un mejor práctica en asignar nombres a las variables son: que inicien con minúscula, que no inicien con un número, pueden contener guión bajo y caracteres alfa numéricos, y finalmente es necesario tomar en cuenta que python distingue las minúsculas y mayúsculas en el nombre de las variables.

En python para saber cuál es el tipo de dato de una variable asignada, se utiliza la función `type()`. Además, puedes forzar la conversión de un dato de tipo carácter con el significado semántico numérico a un dato de tipo numérico y viceversa, con el uso de las funciones de casting `int()`, `float()` y `str()`.

Existen diversas funciones para acceder a la estructura de un dato de tipo cadena. Por ejemplo, para obtener un solo carácter de la cadena “hola mundo”, se puede

declarar una variable con dicha cadena y después llamar a la variable seguida de par corchetes que abren y cierran a la posición numérica, iniciando de 0, del carácter que se necesita. Además, también se pueden generar subcadenas a partir de la misma técnica, pero añadiendo un intervalo de la siguiente manera:

```
subcadena = "cadena"
print(subcadena[1]) # Imprime a
print(subcadena[2:5]) # Imprime dena
```

La función `len()` regresa el largo o el número de caracteres que contiene un cadena. Las funciones `.lower()` y `.upper()` regresan la cadena en minúsculas y mayúsculas respectivamente. Para reemplazar un carácter en una cadena, basta con llamar a la función `.replace()` la cual recibe dos parámetros: el carácter a reemplazar y el carácter con el que va reemplazar. Así pues, se puede dividir una cadena por algún carácter en específico, p. ej., una coma, con la función `.split()`.

El usuario puede interactuar con un programa hecho en python por medio del teclado utilizando la función `input()`, la cual permite asignar un valor de tipo cadena a una variable.

En python existen siete operadores aritméticas: de suma (+), resta (-), multiplicación (*), división (/), exponencial (**), módulo (%) y división de piso (//). A su vez, existen los operadores de comparación como: igual (==), no igual (!=), menor que (<), mayor que (>), menor o igual que (<=) y mayor o igual que (>=). Así como los operadores lógicos: y (and), o (or), no (not).

Una lista se puede definir como el conjunto de valores del mismo tipo de dato separados por comas y encerrada entre corchetes. Además, se pueden agregar valores a la lista a través de la función `.append()` o remover con `.remove()`, entre otras funciones.

Por otro parte, una tupla en python es una colección de valores incambiables, separados por comas y encerrada entre paréntesis. Por lo que, para recorrer los

valores de una tupla, al igual que una lista, un set o un diccionario, se puede hacer uso de estructuras de control de flujo.

Las estructuras de control de flujo sirve para validar condicionantes y promover el flujo del código para el programa. Por lo cual, como se mencionó anteriormente, es importante cuidar la indentación al declararlas.

La condicionante if válida si la condición es verdadera ejecuta lo que está a continuación indentado. Si es falsa, no hará nada y saltará a la siguiente línea de código. También existe la condicionante if else, esto es, si la condicionante es verdadera ejecutará el código correspondiente, sino ejecutará la otra opción seguida de la condición. Lo anterior se muestra en el siguiente ejemplo de elif:

```
a = 400
b = 77

if b > a:
    print("b es mayor que a")
elif a == b:
    print("a y b son iguales")
else:
    print("a es mayor que b")
```

En conclusión, el lenguaje de programación python ha venido a revolucionar la forma de cómo conceptualizamos nuestras expresiones en la interacción humano-máquina. Desde el diseño de lenguajes, representación de algoritmos, traducción del lenguaje natural, entre otros. Por lo que, es válido conocer otras culturas del lenguaje propio por medio del entendimiento que las ciencias de la computación nos brinda.

Webgrafía:

Python tutorial. W3Schools. Recuperado el 26 de mayo de 2019 de de <https://www.w3schools.com/python/>

The Story of Python, by Its Creator, Guido van Rossum. Recuperado el 26 de mayo de 2019 de de <https://www.youtube.com/watch?v=J0Aq44Pze-w>