



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: TISTA GARCÍA EDGAR

Asignatura: ESTRUCTURAS DE DATOS Y ALGORITMOS I

Grupo: 1

No de Práctica(s): 09

Integrante(s): GÓMEZ LUNA ALEJANDRO

*No. de Equipo de
cómputo empleado:* 33

No. de Lista o Brigada: 14

Semestre: 2019-2

Fecha de entrega: 22/Abril/2019

Observaciones:

CALIFICACIÓN: _____

- Objetivo de la práctica
APLICAR LAS BASES DEL LENGUAJE DE PROGRAMACIÓN
PYTHON EN EL AMBIENTE DE IPYTHON NOTEBOOK.

- Desarrollo

Actividad 1

i) En la práctica se nos presentan las bases del lenguaje de programación Python, donde se ejemplifica la forma en como se utilizan las distintas funciones que ofrece el lenguaje.

Se empieza con las variables, las cuales, a diferencia de C, no necesitan llevar el tipo de dato que son, y solo se le asignan el valor que se necesita que almacene. Para poder imprimir el valor que almacenan, se usa la función `print()`, la cual, solo toma como argumento aquello que imprimirá en pantalla, sin tener que especificar formato alguno. También se utiliza la función `type()` para conocer el tipo de dato del cual es la variable, aunque nosotros no declaremos el tipo de dato, el lenguaje lo modifica automáticamente, dependiendo del valor que almacenemos en la variable. En este apartado de variables podemos observar también la existencia de poder declarar cadenas, las cuales tienen un manejo bastante más simple en Python, además de que se ofrecen ciertas operaciones con cadenas, como concatenarlas o cambiar la forma en como se ordenan, esto último haciendo uso de la función `format()`.

Por otro lado, se habla de los operadores aritméticos, de comparación y los booleanos. Cabe resaltar que, los booleanos son un nuevo tipo de operadores que no se encontraban en el lenguaje C.

También se habla de unos nuevos tipos de estructura de datos, los cuales son las listas y las tuplas. La principal diferencia entre estas dos radica en que las listas son mutables, mientras que las tuplas no, esto quiere decir que las listas pueden modificar su valor a través de la ejecución del programa. Las similitudes entre las listas y tuplas es que pueden almacenar distintos tipos de datos, para acceder a los datos que almacenan se usan índices y pueden tener diversas dimensiones. Se puede observar que estos dos tipos de datos funcionan como un tipo de arreglo en el lenguaje C.

Las tuplas también pueden ser «nombradas», con la finalidad de utilizarlas en varias ocasiones para poder crear otra cantidad de tuplas, las cuales tomarán como datos, aquellos que fueron declaradas en la tupla nombrada. Este tipo de dato se puede visualizar como las estructuras en el lenguaje C.

De igual manera, existen los diccionarios, los cuales son un tipo de combinación entre las tuplas y las listas, ya que cuentan con dos tipos de datos, los cuales son llaves y valores, las llaves son inmutables, mientras que los valores si son mutables. Cada llave contiene distintos tipos de valores.

Por último, se habla de las declaraciones de las funciones y las variables de tipo global. Las funciones utilizan la palabra reservada **def** para definir las, y al igual que las variables, no se tiene que especificar el tipo de dato que retornan o el tipo de dato que toman como parámetros. Todas las variables creadas dentro de cualquier función son de tipo local y aquellas que son declaradas afuera de cualquier función, son de tipo global, por lo que pueden ser ocupadas en cualquier lugar del programa.

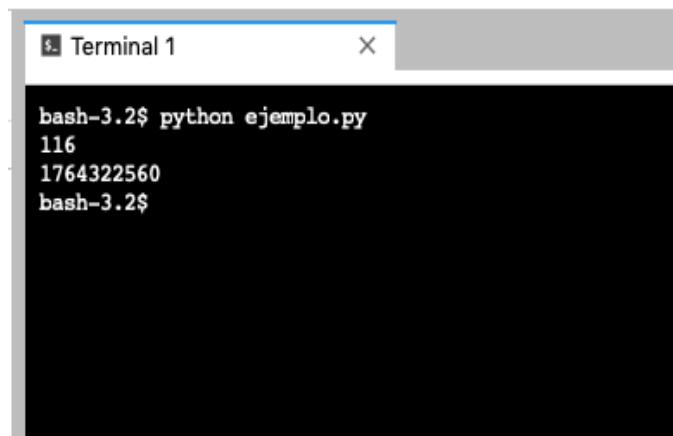
ii) Para este programa, en las dos funciones, se toma como parámetro el valor que se va a convertir para posteriormente almacenarlo en una variable, aplicando la respectiva operación aritmética a la conversión. Por último, se imprime este valor en pantalla. Para comprobarlo, se usa a la función creada en su respectiva celda.

```
In [7]: def transformarMillas(millas):  
        x=millas*1.60934  
        print("Distancia en kilometros es: ",x)  
  
        transformarMillas(5)  
  
        Distancia en kilometros es: 8.0467
```

```
In [8]: def transformarKilometros(kilometros):  
        x=kilometros/1.60934  
        print("Distancia en millas es: ",x)  
  
        transformarKilometros(8.0467)  
  
        Distancia en millas es: 5.0
```

Actividad 2

a) La salida del programa fue la siguiente:



```
Terminal 1  
bash-3.2$ python ejemplo.py  
116  
1764322560  
bash-3.2$
```

b) Como podemos observar en la salida, se muestran dos números, cada uno corresponde al valor resultante en cada una de las variables después de aplicar los respectivos ciclos for. Para dichos ciclos, se utilizan los valores con los que fue creada la lista. Estos valores se utilizan ya que, el ciclo empieza desde el primer valor en la lista y se repite hasta recorrer todos los valores que hay en ella.

La diferencia entre los dos ciclos es que, a la variable valor1, se le van sumando los valores de la lista, mientras que, a la variable valor2, se le van multiplicando los valores de la lista.