	Carátula para entrega de prácticas
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: M.I. Edgar Tista García

Asignatura: Estructura de Datos y Algoritmos I

Grupo: 1

No de Práctica(s): 9

Integrante(s): Díaz Hernández Marcos Bryan

No. de Equipo de cómputo empleado:

No. de Lista o Brigada: 9

Semestre: 2020-2

Fecha de entrega: 26 de marzo de 2020

Observaciones:

CALIFICACIÓN: _____

Objetivo de la práctica

Aplicar las bases del lenguaje de programación Python en el ambiente de ipython notebook.

Introducción

El reporte siguiente me ha gustado hacerlo, por ello he explicado los términos y lo realizado en cada ejercicio, de la forma más clara posible, en base al contenido teórico que envuelve a la práctica.

Ejercicios de la guía de laboratorio – Actividad 1

- Ejercicio 1

El primer ejercicio consistía en ejecutar las celdas de código en el Note Book, para ello se podía ejecutar por celda individual o correr todas las celdas, para ver que hacían, fui ejecutando cada celda y ver los resultados. Por ello esta parte consiste en el análisis de ciertas partes que me llamaron la atención.

Algo que me llamó la atención fue el código del programa correspondiente a la práctica, debido a que lucia bastante desastroso, pero al ejecutarlo en el Note Book, se veía muy bien.

En la imagen 1, hubo un error con el tipo de dato debido a que no se idéntico a “cadena”, a pesar de que estuviera definida en la parte superior del código.

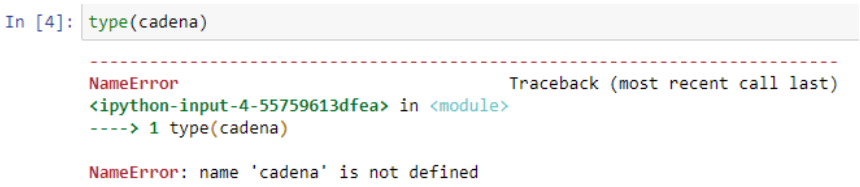


Imagen 1

Como en la clase en línea se decía, existen los valores mutables y no mutables, en este caso se crea una tupla, a la que se le asigna un nombre y se intenta modificar. Dando como resultado un error en el código, esto porque el contenido de una tupla no es modificable (Imagen 2).

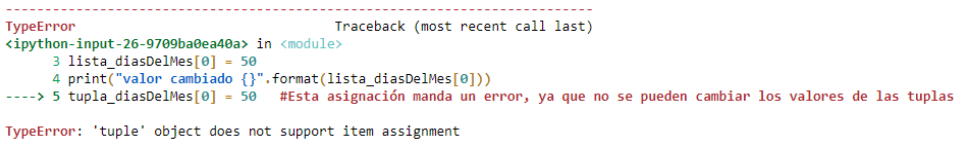


Imagen 2

En la imagen 3, se muestra un error, debido a funciones y a los límites de las variables cuando son definidas dentro de una función, demostrando que, si se intenta utilizar una variable que no esta definida dentro de una función, el resultado será un error. Además de ser un mal uso de las variables.

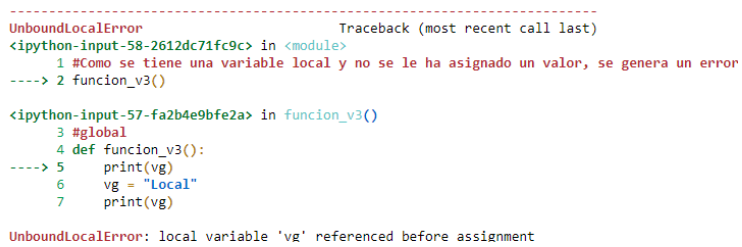


Imagen 3

Un tipo de resumen de mi primer experiencia con Python, fue que me sentí algo perturbado por la libertad que da al programador, y por que rompe esquemas de otros lenguajes de programación, lo que me agrado bastante, e implica mucha creatividad para poder resolver un problema, y que es necesario como en C tener las consideraciones de lo que se planea hacer, pero podría decirse que Python es mas general que C, ya que por ejemplo permite devolver mas de un valor de una función, crear estructuras de forma simple, combinar operadores lógicos, algo que muchas veces facilita el proceso de codificación.

- Ejercicio 2

La segunda parte de la primera actividad fue más práctica, debido a que esta parte consistía en elaborar un programa por medio de celdas en el Note Book, se pedía utilizar funciones y colocar cada función dentro de una celda distinta. El programa convierte de grados Celsius a Fahrenheit y viceversa, también convierte de Millas a Kilómetros y viceversa.

Una parte que me dio bastantes dudas fue si el programa debía tener un valor y modificarlo, o debía ser insertado un valor por el usuario para ser modificado, siendo la segunda forma la que escogí para hacer el código.

- Dificultades en el código

La primera actividad fue de mucha ayuda porque pude identificar el cómo, imprimir valores, guardar valores, utilizar funciones, en Python. Lo más difícil fue el buscar la sintaxis de Python para lo que yo pensaba hacer, entonces me puse a buscar información para crear un tipo case, porque no encontraba otra forma de plantear el ejercicio.

Otro problema que tuve fue el guardado de valores que insertaba el usuario, porque la instrucción de código que encontré guardaba los valores, pero como cadenas, es decir tomaba los valores con caracteres y me causaba problemas cuando se realizaban las operaciones (Imagen 4).

```
b=input("Inserta la opcion:") #Inserta caracter
a=float(input("Inserta el valor:"))
```

Imagen 4

- ¿Cómo lo resolví?

Al buscar no encontré algo similar al case en C, aun así, encontré algunas instrucciones que sirvieron para definir una comparación entre valores que insertaba el usuario y las operaciones que se llevarían a cabo si se cumple la condición.

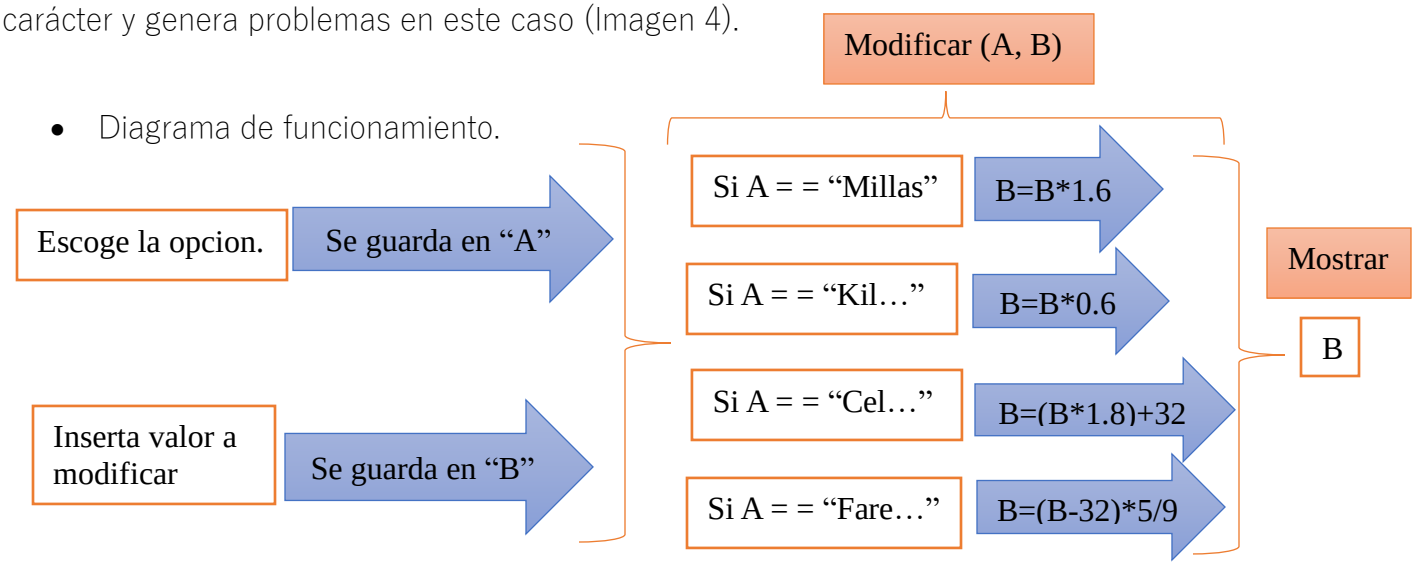
Con un operador de comparación "elif", se me ilumino el mundo, debido a que este operador sirve para poder llevar a cabo una comparación y su caso contrario al mismo tiempo, lo que yo hice fue usarla para el "else" de mi primer "if", en la función que modifica los valores (Imagen 5).

```
elif y=='F':
    x=(x-32)*5/9
    print("Valor en Centigrados: {}".format(x))
```

Imagen 5

Para resolver la entrada de datos por parte del usuario, es necesario realizar un tipo de casteo para definir el tipo de dato que el usuario va a insertar, debido a que, si se deja la instrucción "input", la instrucción que recibe un valor del usuario, todo lo que se inserte se identifica como una cadena o un carácter y genera problemas en este caso (Imagen 4).

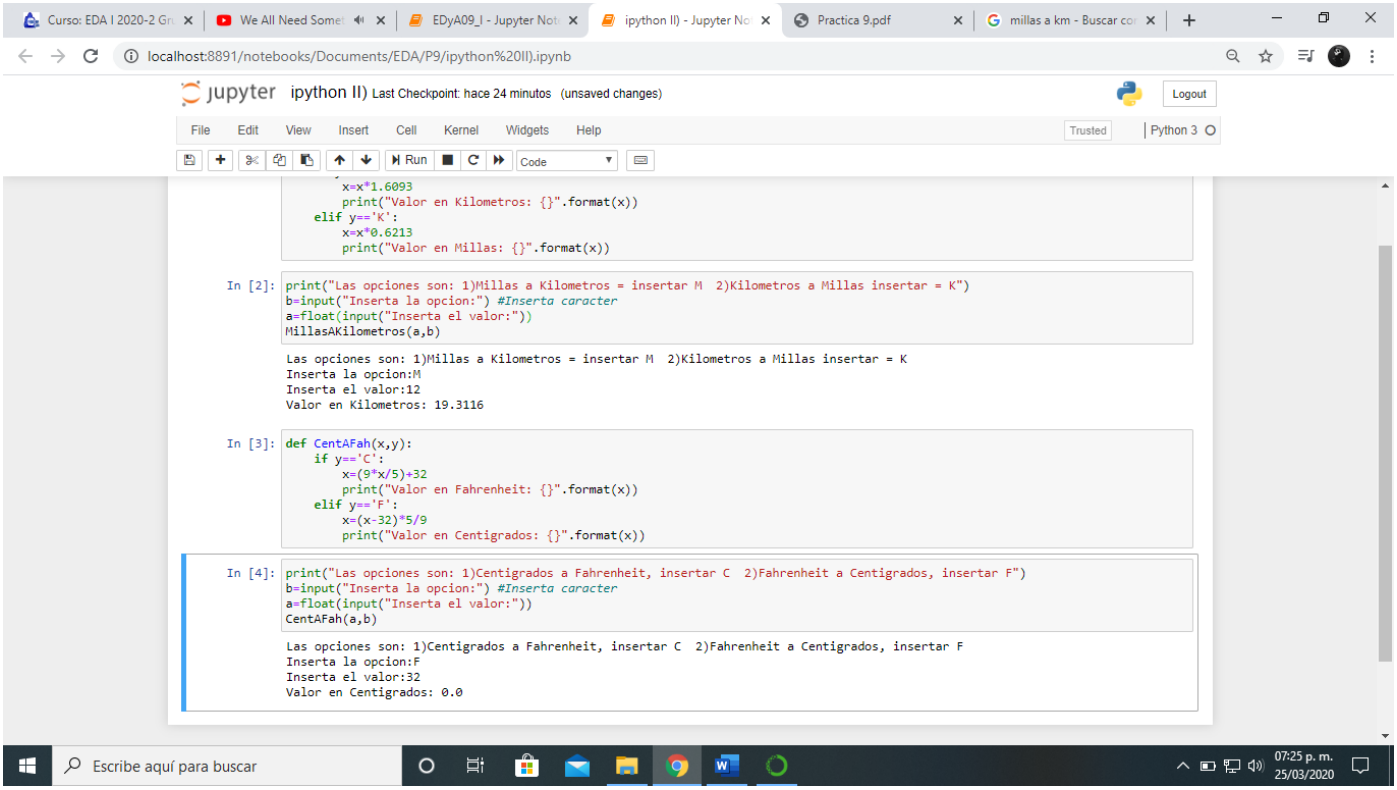
- Diagrama de funcionamiento.



- Relación con teoría.

Con la teoría vista en línea, tiene relación, pero tiene más relación con el primer ejemplo de la práctica, porque muestra de forma explicita como se usan las variables, como se modifican, es mas visual y es de mayor ayuda. No obstante, es prácticamente el uso de los conocimientos básicos de Python para poder resolver el ejercicio.

- Evidencia de Implementación.



Actividad 2

- Ejercicio 1

El ejercicio consistía en crear un documento con terminación (.py) en el cual se escribió un código y por medio de símbolo el sistema, ejecutar el archivo con el comando Python Archivo.py. Además de comprobar la instalación correcta de Python por medio de la verificación de la versión.

La salida en el símbolo de sistema no era igual a la que el profesor tenía en el video, sino que fue la siguiente, donde se muestra la versión, además de un mensaje del ambiente de Anaconda que no esta activado (Imagen 1).

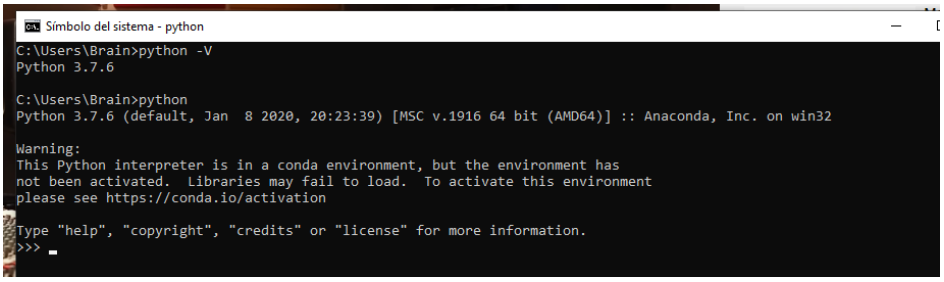


Imagen 1

- Dificultades

La mayor dificultad que tuve fue el ejecutar el archivo que donde guarde el código, debido a la terminación que indica el tipo de archivo. Lo que sucedió que fue la terminación del archivo era (.txt), por lo que al buscar el archivo con (.py), no lo encontraba el símbolo del sistema (Imagen 2).

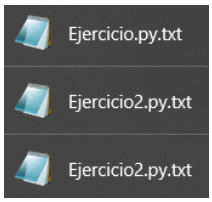


Imagen 2

Además de que era necesario el indicar la ubicación casi exacta del archivo, ya que Python no busca dentro de las carpetas, sino que lo hace por encima (Imagen 3).

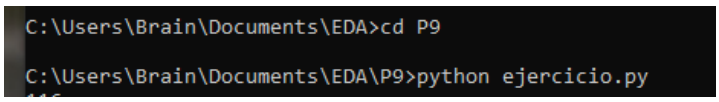


Imagen 3

- Indica la salida del sistema

Las salidas del sistema corresponden a Valor1=116 y Valor2=1764322560 (Imagen 4).

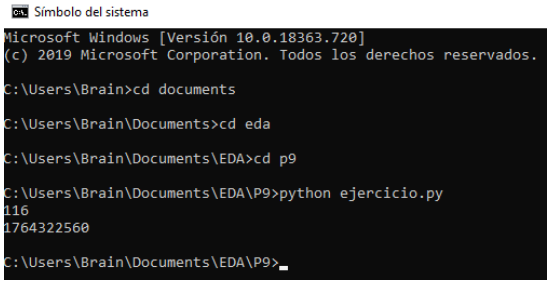
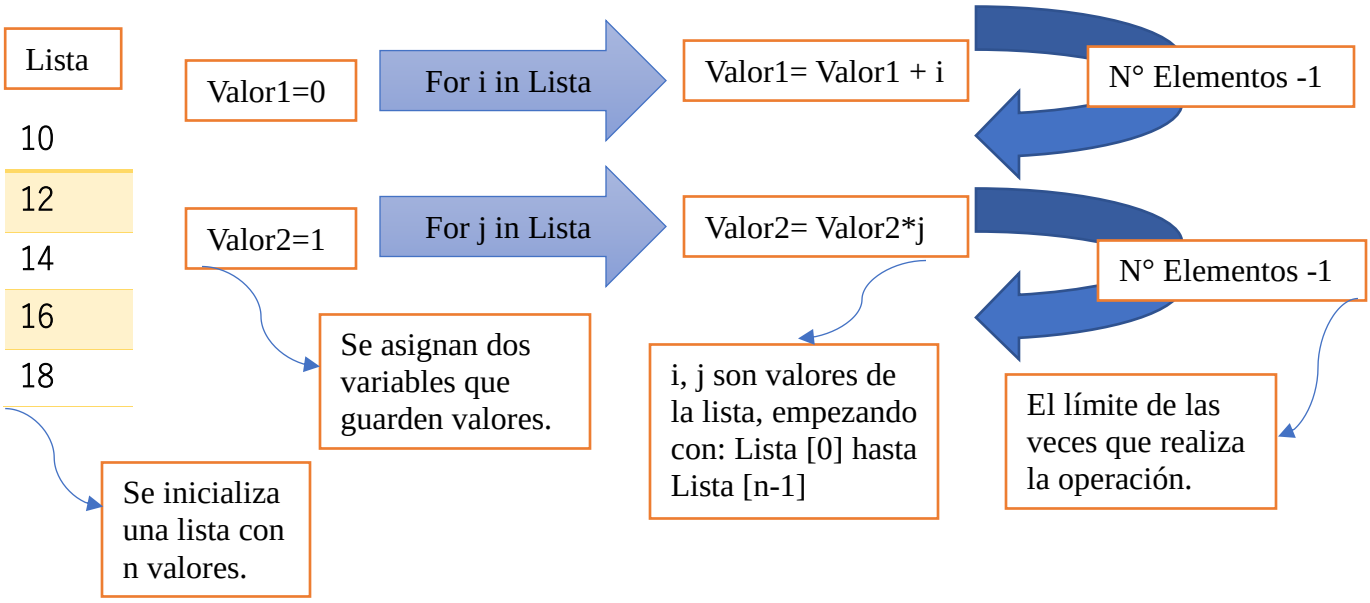


Imagen 4

- Explica que hace el programa

El programa lleva a cabo dos ciclos for, el primero de los ciclos son sumas sobre un mismo número, es decir que se realiza la suma de un elemento nuevo, a un elemento que se suma a si mismo más un incremento, esto lo realiza n veces en base al ciclo. El segundo ciclo lleva a cabo multiplicaciones sobre un mismo elemento, que multiplica con un nuevo valor n veces. Los valores que suma y multiplica cada for, son los valores que pertenecen a una lista definida en el código.

- Diagrama de funcionamiento.



Actividad 3

- Diferencias Jupyter Notebook y ejecucion desde consola.

Con Jupiter se me hizo mas sencillo el poder programar, debido a que la interfaz indica donde estan los errores, cuales son los errores, permite poder hacer mas modificaciones de los archivos de los programas y del propio código. Funciona bastante bien si se esta aprendiendo a programar en Python, porque equivocarse y detectar el error es rápido, además de que se puede corregir en el momento.

En el caso contrario la interfaz requiere mayor cuidado y experiencia, porque el hacer el código en un archivo, y que tenga fallas no es algo que se eficiente, además que se puede volver en algo molesto y frustante.

- Mas facil para trabajar y ¿porque?.

Al nivel que estoy, la verdad trabajar en Jupiter es mejor, porque permite el poder aprender por medio de errores y acierto, además de que es ciertas cosas es intuitivo y funciona bien para alguien que esta comenzando a programar en Python.

Conclusiones.

El comenzar un nuevo lenguaje siempre trae incertidumbre porque a veces puede ser algo totalmente distinto a lo que conocemos, aun así siento que he aprendido bastante del lenguaje en poco tiempo y me ha dejado los conocimientos basicos para poder realizar un programa. Para concluir, he logrado realizar un programa utilizando conceptos basicos y tambien algunos extra, para poder aprender los necesario y poder realizar un programa en Python.