Estructura de Datos y Algoritmos I **25/07/2021**

Tarea especial Python

Python es un lenguaje de programación de alto nivel, tiene un propósito general (se trabaja sobre diferentes temas), es multi-paradigma (manera estructurada, funcional, orientado a objetos, reflexivo, etc.) y es interpretado. Fue inventado por Guido van Rossum a finales de los años ochentas.

Python tiene una colección de 20 principios de software llamados **Zen de Python**, los cuales influyen en el diseño del lenguaje de programación, los primeros 4 principios son:

1. *Bonito es mejor que feo.*
2. *Explícito es mejor que implícito.*
3. *Simple es mejor que complejo.*
4. *Complejo es mejor que complicado.*

Práctica #9

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

Desarrollo

Actividad 1

El lenguaje de Python se caracteriza por ser distinto al lenguaje C debido a cómo se escribe ya que este no se requiere especificar los tipos de datos a utilizar, funciones, estructuras, etc. Esto hace que escribir el código más sencillo, se utilizan menos líneas de código a comparación del lenguaje C y para el humano, es más comprensible. De las cosas que más me impresionó Python a comparación de C son:

- Se puede asignar un mismo valor a varias variables al mismo tiempo.
- Las variables pueden cambiar de valor sin importar el tipo de dato que contengan.
- No se requiere el uso de punto y coma “;” para las líneas de código ni de llaves “{}” para las funciones.
- No es necesario declarar previamente las funciones y ni de qué tipo de función son.
- Se utiliza tabuladores (normalmente de 4 espacios) para diferencia si este se encuentra dentro de un bloque de código.
- Existen las tuplas y diccionarios.
- Para modificar una variable global dentro de una función se utiliza la palabra reservada **global**.

Desde mi punto de vista, el lenguaje Python es más fácil de usar y de comprender que el lenguaje C.

Actividad 1.2

Para esta actividad, se creó un archivo en la notebook con el nombre "*Ejercicio 2 PR9.ipynb*" en el cual, contiene 4 celdas con 4 funciones distintas, las cuales son:

- *piesAMetros():* La función recibe como parámetro el valor en pies y retorna a través de una multiplicación el valor en metros.
- *metrosAPies():* La función recibe como parámetro el valor en metros y retorna a través de una multiplicación el valor en pies.
- *centigradosAFahrenheit():* La función recibe como parámetro el valor en centígrados y retorna a través de una multiplicación el valor en Fahrenheit.
- *fahrenheitACentigrados():* La función recibe como parámetro el valor en Fahrenheit y retorna a través de una multiplicación el valor en centígrados.

Para terminar, en la última celda se imprime los valores que retorna cada función al pasarle un parámetro, el resultado está redondeado a solo imprimir los 2 primeros decimales y en este caso se escogió como parámetro el valor de **15.**

```
Se utiliza y se pasa como parametro el valor de 15 para cada funcion
Metros: 4.57
Pies: 49.21
Fahrenheit: 59.0
Centigrados: -9.44
```

Actividad 2

El programa llamado "*actividad2.py*", utiliza una lista de números enteros y dos variables. Se utilizan dos *ciclos for* por separado para que se recorra los elementos de la lista.

- El primer *ciclo for* sumará los elementos de la lista y lo guardará en una variable
- El segundo *ciclo for* multiplicará los elementos de la lista y lo guarda en una variable

```
68
19958400
```

Al final se imprime el valor de la suma y de la multiplicación.

Por último, el programa se puede mejorar al utilizar solo un *ciclo for*, ya que solo varía la operación a realizar, esto queda de la siguiente manera:

```
for i in lista:
    valor1 += i
    valor2 *= i
```

Actividad 3

Las principales diferencias al ejecutar archivos entre Jupyter Notebook y la consola, son:

En Jupyter:

- Se ejecuta e imprime (si fuera el caso) la información del código a través de celdas.
- Se puede decidir qué celdas ejecutar y en qué orden.
- En las celdas, se puede implementar funciones y ser utilizadas en celdas posteriores.

- Se ejecuta y muestran de mejor manera las gráficas.
- Se puede dividir en secciones el programa al agregar celdas de texto, esto facilita la comprensión del programa. La celda del texto no se imprime.
  En consola:
- No se imprime la información como Jupyter Notebook.
- La información impresa sigue un orden de manera que el programador la implemente.
- El programador debe de implementar maneras para que a información se vea de mejor manera.

Además, desde mi punto de vista, es más agradable utilizar Jupyter Notebook debido a que la interfaz del usuario esta mejor, se puede implementar de mejor manera el uso de librerías y en especial, las librerías para graficar. En mi opinión, al trabajar con celdas es trabajar para "dividir y conquistar" el programa y esto provoca un mejor control de flujo y al utilizarlas es más fácil encontrar errores,

Conclusiones

El lenguaje Python tiene muchas opciones de trabajo, pues se puede trabajar con las variables, funciones, estructuras lineales de manera más sencilla que en el lenguaje C reduciendo la cantidad de líneas del código y estrés para el programador y usuario.
No se me complicó entender el lenguaje, comprender los programas propuestos por el profesor ni crear un programa para la actividad 2, me gusta el lenguaje Python.
Los programas cumplieron los puntos propuestos por el profesor.
Los objetivos de la práctica fueron cumplidos en su totalidad al aplicar las bases del lenguaje en los programas.

Práctica #10

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

Desarrollo

Actividad 1

Hay muchas diferencias entre el lenguaje Python y el lenguaje C, pero lo que más me llamó la atención de Python fue que:

- En el caso de las estructuras de control selectivas, lo impresionante es que ya no se usan las llaves "{}" para utilizar las *sentencias if*, solo se utiliza los dos puntos ":" y dentro de la estructura se debe de ocupar un tabulador de espacio, además, en el caso de comprobar con un *if* si 3 variables son iguales es mucho más sencillo que en lenguaje C.
- En el caso de las estructuras de control repetitivas, me di cuenta de que no existe el *ciclo Do-While* en Python, sin embargo, en el *ciclo for* lo que más me sorprendió fue que hay más maneras de emplearse pues se puede recorrer una lista, un diccionario

(key, value o ambos) un rango entre números, una cadena de texto y puede haber más de 1 variable que recorre la lista, esta puede ser de tipo **int** o **str**.

- Las bibliotecas entre Python y C tienen distintos objetivos, en el caso de Python hay varias librerías para funciones matemáticas, para graficar, para el aprendizaje de una máquina, para la manipulación de datos de las hojas de cálculos, entre otras.
- Python puede graficar en 2D y en 3D, se le puede modificar el color de la gráfica, el título, los nombres de los ejes, tipo de gráfica y la podemos guardar en una imagen.
- Para que el usuario ingrese datos solo es necesario usar la función *input()*¸escribir la instrucción dentro de la función y asignar la función a una variable, esta puede guardar cualquier tipo de dato y no es necesario especificar el tipo de dato.

Actividad 2

Se trabajó en el IDE "Spyder", se creó dos programas llamados "*contraseña2-1.py*" y "*contraseña2-2.py*". En el primer programa, se creó una constante con el nombre **contraseña** y se le solicita al usuario que ingrese el valor correcto de la **contraseña**, no acepta otro valor distinto a la contraseña, por eso, se le solicitará el valor de esta hasta que ingrese el valor correcto. En el segundo programa se le implementó un número de intentos (4) y si ingresa el valor correcto, se imprime un mensaje de bienvenida, si se le acaban los intentos, se finaliza la ejecución del programa.

En el primer programa, a través de un *ciclo while* se le solicita al usuario que ingrese la contraseña, no tiene un límite de intentos ya que el ciclo se repetirá hasta que el ingrese la contraseña correcta.

En el segundo programa, para que se le solicite al usuario que ingrese la contraseña teniendo 4 intentos, se utilizó una variable **i** con valor de **1**, un *ciclo while* que se ejecutará siempre y cuando el valor de **i** sea distinto de **5**. Si la contraseña ingresada es la misma contraseña almacenada, se rompe el *ciclo while* y se le da la bienvenida al usuario, de lo contrario, se le indica cuantos intentos le queda, se aumenta una unidad el valor de **i** y si se le acaban las oportunidades, se le notifica al usuario que se acabaron sus intentos.

```
Ingrese la contraseña: 72171
Le queda 3 intentos

Ingrese la contraseña: duqbdqebd
Le queda 2 intentos

Ingrese la contraseña: 28eh18eh
Le queda 1 intentos

Ingrese la contraseña: du9qdb12d
Le queda 0 intentos
Se acabaron los intentos
```

```
Ingrese la contraseña: uhf8hf2
Le queda 3 intentos

Ingrese la contraseña: Kf#23dh-3
Bienvenido
```

La contraseña es: Kf#23dh-3

Actividad 3

Se creó un programa un programa llamado "*calificaciones.py*" en donde se capturan las calificaciones de tareas, exámenes y prácticas de laboratorio, se obtiene el promedio de cada

actividad y el promedio final. Por último, conforme al promedio de la tarea se va a modificar el promedio del examen y esto altera el promedio final.

El programa inicia en “if __name__ == ‘__main__’ ” en donde se llama la función *run()*, el cual se encuentra toda la estructura del programa.
Se le solicita al usuario cuántas calificaciones se va a ingresar de tareas, de exámenes y de prácticas, cada valor se guarda en variables distintas.

```
¿Cuantas calificaciones de tareas va ingresar? 3
¿Cuantas calificaciones de examenes va ingresar? 4
¿Cuantas calificaciones de prácticas de laboratorio va ingresar? 3
```

Se utilizan 3 *ciclos for*, cada *ciclo for* se va a ejecutar el número calificaciones a ingresar conforme a su actividad (tareas, exámenes, practicas) . Dentro del *ciclo for* se le solicita al usuario que ingrese la calificación y esta se va a guardar en una lista (cada actividad tiene su propia lista)

```
Ingrese la calficación de la tarea #1 9.5
Ingrese la calficación de la tarea #2 9.5
Ingrese la calficación de la tarea #3 9.5
```

```
Ingrese la calficación del examen #1 8.5
Ingrese la calficación del examen #2 8.5
Ingrese la calficación del examen #3 8.5
Ingrese la calficación del examen #4 8.5
```

```
Ingrese la calficación de la práctica #1 10
Ingrese la calficación de la práctica #2 10
Ingrese la calficación de la práctica #3 10
```

Con otros 3 *ciclos for* (cada uno corresponde a la actividad) se obtendrá el promedio.

Una vez obtenido el promedio de los exámenes y de las prácticas, al tener una evaluación dividida en exámenes con valor de 65% y prácticas con valor 35%, para obtener dichos puntos, el promedio del examen es multiplicado por 0.65 y el promedio de las prácticas es multiplicado por 0.35. Estos se suman y se obtiene la calificación final de la materia.
Se imprime el promedio del examen, de laboratorio y la calificación final obtenida.

Con una *sentencia if* se valida el promedio de la tarea, si este es mayor a 8.5 se agrega **0.5** al promedio de los exámenes, si es menor que 8.5 pero mayor de 7 no se hace nada y si es menor de 7.5 se resta **0.5** al promedio de los exámenes.
Se imprime el promedio de las tareas y la calificación final modificada.

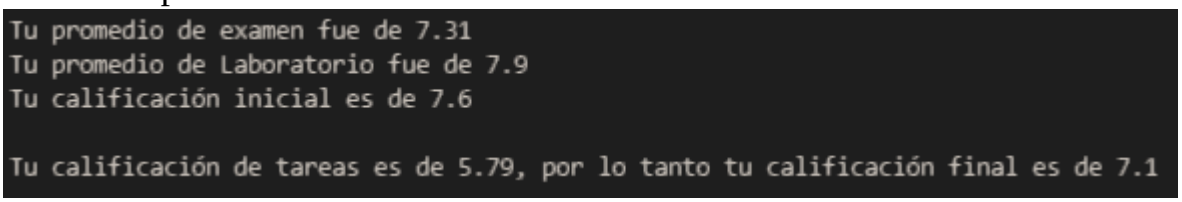
Tarea con promedio mayor a 8.5:

```
Tu promedio de examen fue de 8.5
Tu promedio de Laboratorio fue de 10.0
Tu calificación inicial es de 9.0

Tu calificación de tareas es de 9.5, por lo tanto tu calificación final es de 9.5
```

Tarea con promedio entre 7.0 y 8.5:

```
Tu promedio de examen fue de 8.0
Tu promedio de Laboratorio fue de 9.0
Tu calificación inicial es de 8.3

Tu calificación de tareas es de 7.67, por lo tanto tu calificación final es de 8.3
```

Tarea con promedio menor a 7.0:



Conclusiones

En un principio, tuve problemas en la actividad 3 al querer realizar operaciones y utilizar los *ciclos for* con una variable de tipo **int** y otra de tipo **string,** esto se arregló al cambiar la variable de tipo **string** a **int** con la función *int()*.

El IDE Spyder me gustó más que la consola por su fácil manejo e interfaz de usuario, además tiene mejor eficacia para ejecutar los programas.

Los programas cumplen los objetivos de cada actividad, siendo esto la realización de un 100%.
Se implementaron las bases del lenguaje, por esto, se cumplen los objetivos de la práctica.