|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B



|  |  |
| --- | --- |
| Profesor(a): | M.I. Marco Antonio Martinez Quintana. |
| Asignatura: | Estructura de Datos y Algoritmos I |
| Grupo: | 17 |
| No de Práctica(s): | 10 |
| Integrante(s): | Pineda González Rodrigo |
| No. de Equipo de cómputo empleado: | 26 |
| No. de Lista o Brigada: | 30 |
| Semestre: | 2020-II |
| Fecha de entrega: | 14/04/2020 |
| Observaciones: |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_

# Objetivo:

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

# Introducción:

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad. El creador del lenguaje es Guido Van Rossum. Hace ya más de una década que diseñó Python. El objetivo era cubrir la necesidad de un lenguaje orientado a objetos de sencillo uso que sirviese para tratar diversas tareas dentro de la programación que habitualmente se usando C.

Dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de cadenas, números, archivos, entre otros. Además, existen muchas librerías o bibliotecas, que se encuentran en la colección de “The Python Standard Library”, la mayoría multi-plataforma, que podemos importar para tratar temas específicos como la programación de ventanas o sistemas en red. Para separar las porciones de código en Python se debe tabular, colocando un margen al código que irá dentro de una función o un bucle.

## Bibliotecas

Algunas de las bibliotecas más utilizadas son:

* NumPy (Numerical Python). Es usado para realizar operaciones con vectores o matrices de una manera eficiente. Contiene funciones de Álgebra Lineal, transformadas de Fourier, generación de números aleatorios e integración con Fortran, C y C++.
* SciPy (Scientific Python). Hace uso de Numpy y es utilizada para hacer operaciones más avanzadas como transformadas discretas de Fourier, Álgebra Lineal, Optimización, etc.
* Matplotlib. Esta biblioteca es usada para generar una variedad de gráficas en 2D y 3D.
* Scikit Learn (Machine Learning). Esta biblioteca está basada en los anteriores y contiene algoritmos de aprendizaje de máquina, reconocimiento de patrones y estadísticas para realizar clasificación, regresión, entre otros.
* Pandas (Manipulación de datos). Esta biblioteca es utilizada para manipulación de datos, contiene estructuras de datos llamadas data frames que se asemejan a las hojas de cálculo y a los cuales se le puede aplicar una gran cantidad de funciones.

## Graficación

Matplotlib es una biblioteca usada para generar gráficas en 2D y 3D, donde cada una de las configuraciones de la gráfica es programable.

## Ejecución desde ventana de comandos

Al igual que en otros lenguajes, también se puede se le puede pedir al usuario que introduzca ciertos datos de entrada cuando se ejecute un programa. Esto no se puede hacer desde Jupyter notebook, ya que los datos se introducen en las celdas que se van agregando a lo largo de la página.

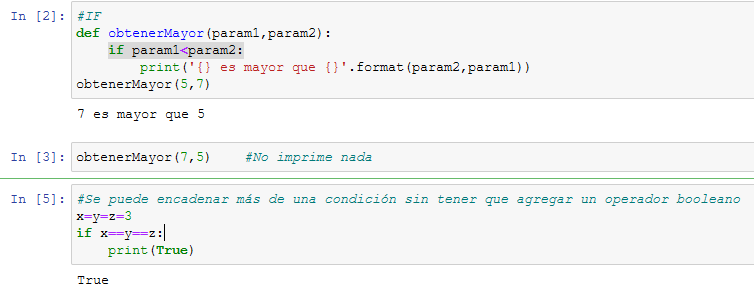
## Repositorio de la práctica:

<https://github.com/RodrigoPG-W/Python10EDAI>

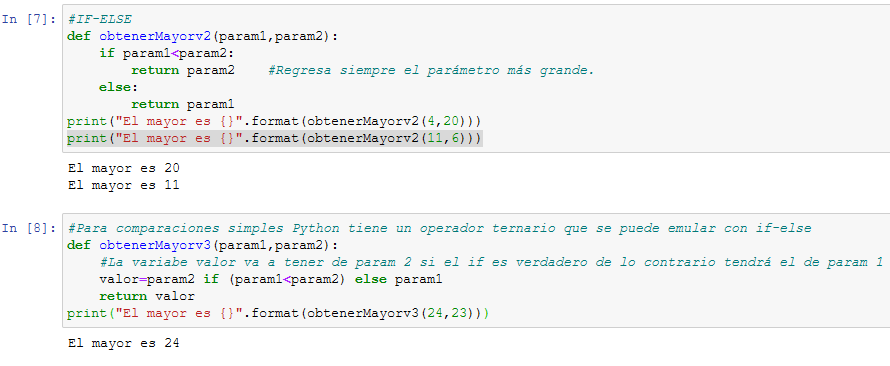
# Desarrollo:

## Estructuras de control selectivas

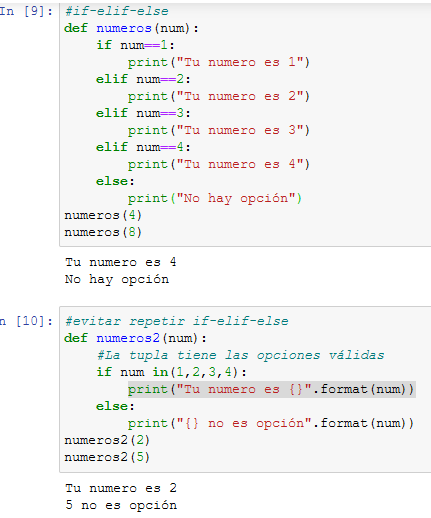
* IF: sirve para ejecutar código dependiendo del resultado de una condición.

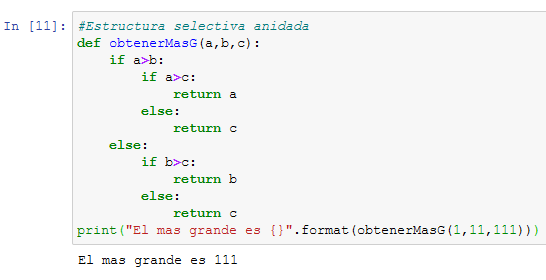


* IF-ELSE: Este tipo de declaraciones se usan para dar una opción en el caso de que la condición no se cumpla.



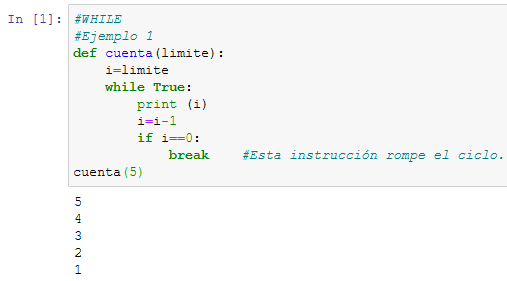
* IF-ELIF-ELSE: sirve para generar varias casos de prueba. similar a la instrucción switch del lenguaje C.

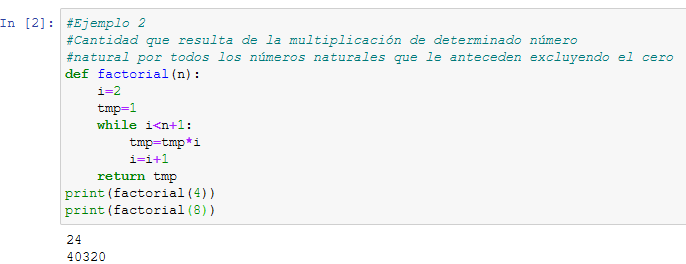




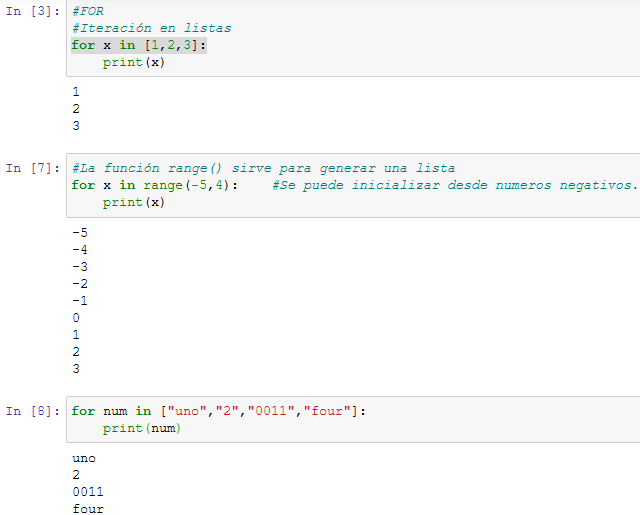
## Estructuras de control repetitivas

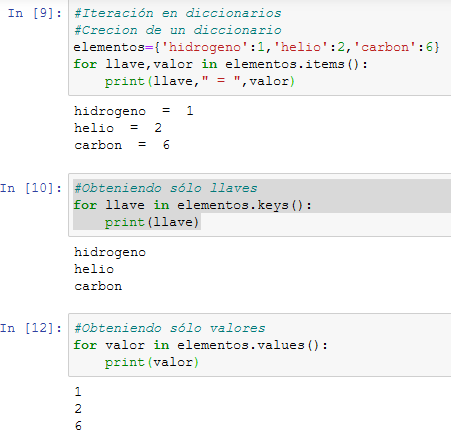
* WHILE: Para que el ciclo se ejecute, la condición siempre tiene que ser verdadera.

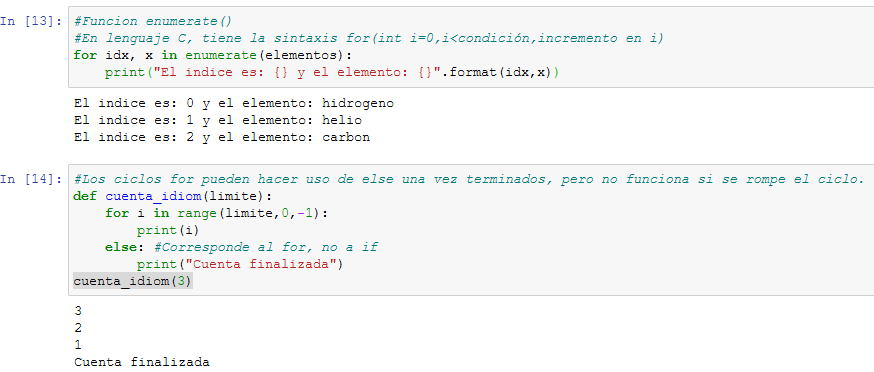


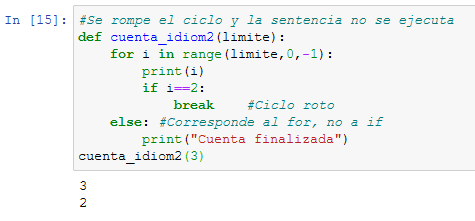


* FOR: se utiliza generalmente para hacer iteraciones en una lista, diccionarios y arreglos.

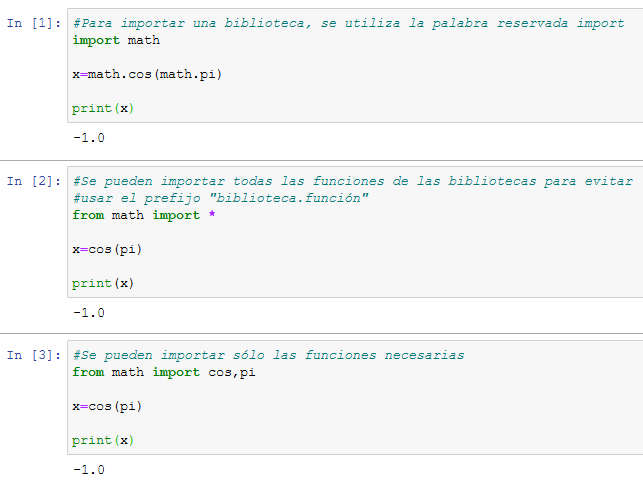


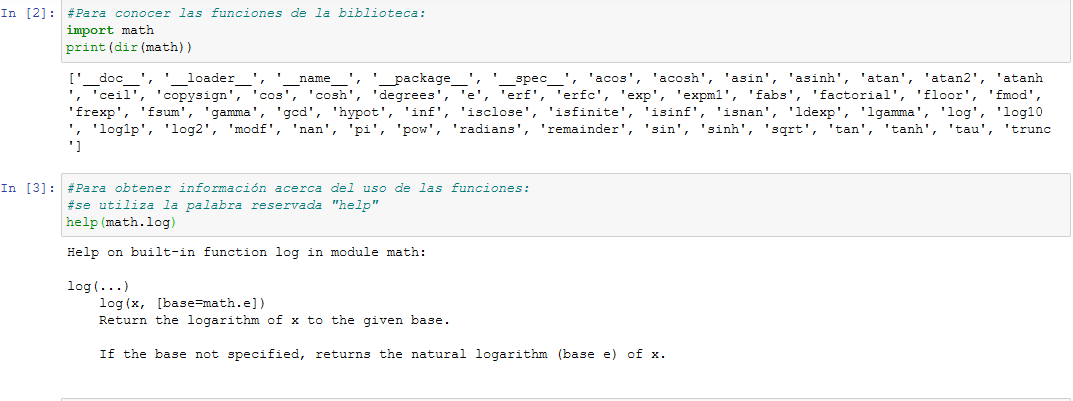


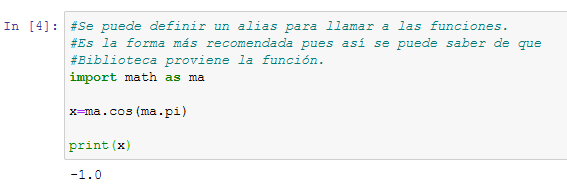




## Bibliotecas



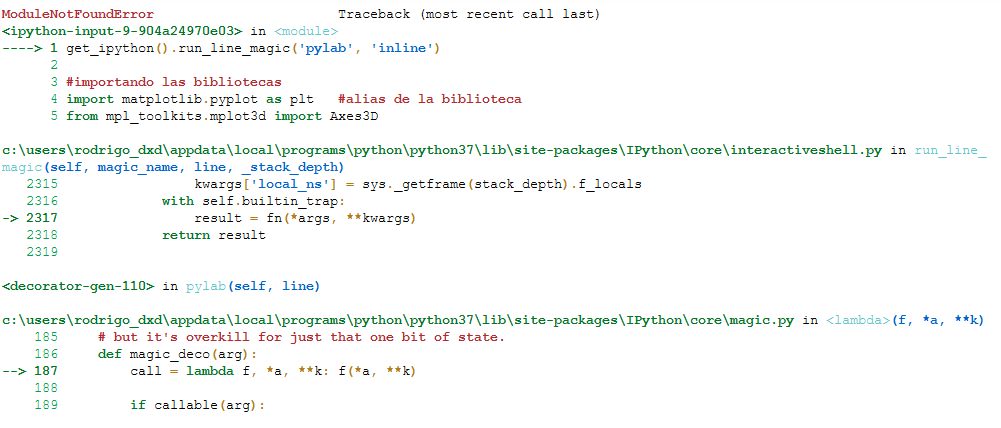


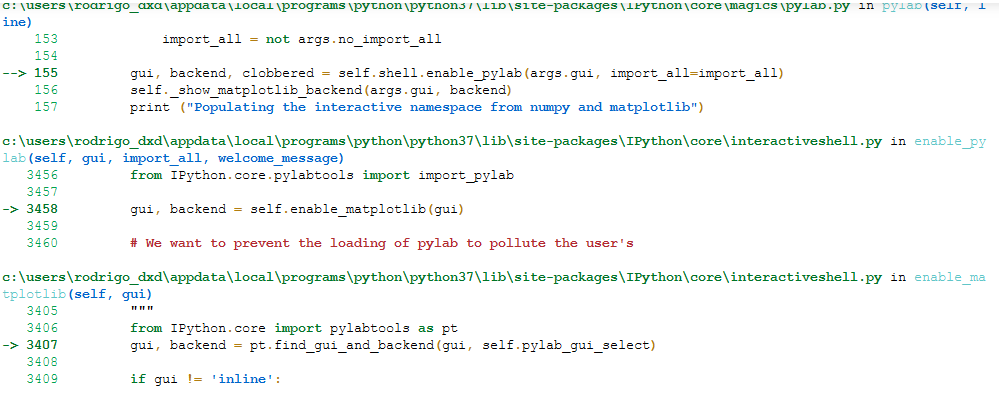


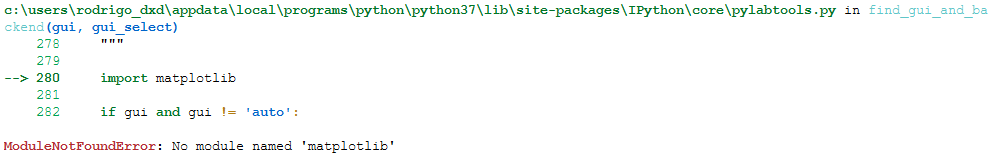
## Graficación



No se pudo ejecutar el programa contenido en la práctica, debido a los siguientes errores:







# Conclusión:

El uso del lenguaje Python se ve bastante favorecido con el apoyo que cuenta, pues se han desarrollado bastantes librerías y el soporte que hay para aprender es muy grande. Debido a la sencillez en cuanto la especificación de tipos, acciones y sintaxis, lo hace un lenguaje más fácil para trabajar y cometer menos errores sintácticos. A su vez, y siendo uno de los lenguajes más prometedores y utilizados en la actualidad, el conocerlo resulta una gran herramienta para el mundo laboral. Sin embargo, considero que, a pesar de ser muy útil gracias a su portabilidad a otras plataformas, hay soluciones que pueden realizarse con lenguajes de programación que permitan optimizar los tiempos y el manejo de memoria de mejor manera. Respecto al entorno Jupyter, no me resultó demasiado agradable, sin embargo, es funcional para propósitos didácticos. Otra crítica respecto a esta práctica es que no viene demasiado claro el como realizar algunos ejercicios correctamente, causando que los programas no funcionen.

Referencias:

* Miguel Angel Alvarez (2003), “*Qué es Python*” , (consultado el 14/04/2020). Recuperado de: <https://desarrolloweb.com/articulos/1325.php>
* Tutorial oficial de Python: <https://docs.python.org/3/tutorial/>
* Galería de notebooks: <https://wakari.io/gallery>
* Matplotlib: <http://matplotlib.org>