|  |  |  |
| --- | --- | --- |
|  | **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): | 09 |
| Integrante(s): | Pineda González Rodrigo |
| No. de Equipo de cómputo empleado: | 26 |
| No. de Lista o Brigada: | 30 |
| Semestre: | 2020-II |
| Fecha de entrega: | 31/03/2020 |
| Observaciones: |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_

# Objetivo:

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

# Introducción:

Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos.El intérprete de Python y la extensa biblioteca estándar están a libre disposición en forma binaria y de código fuente para las principales plataformas desde el sitio web de Python,<https://www.python.org/>, y puede distribuirse libremente. El mismo sitio contiene también distribuciones y enlaces de muchos módulos libres de Python de terceros, programas y herramientas, y documentación adicional.

El intérprete de Python puede extenderse fácilmente con nuevas funcionalidades y tipos de datos implementados en C o C++ (u otros lenguajes accesibles desde C). Python también puede usarse como un lenguaje de extensiones para aplicaciones personalizables.

## Variables y tipos

Son similares a los de lenguaje C, con la ventaja de no hay que especificar el tipo pues ya viene implícito. Otras consideraciones importantes son:

* Los nombres de las variables son alfanuméricos (a-z, A-Z, 0-9) y empiezan con una letra en minúscula.
* No se necesita poner ; al final de cada instrucción.
* Mantener las indentaciones al momento de escribir código. Se recomienda usar 4 espacios por nivel de indentación, los espacios son preferidos sobre el uso de tabuladores.

Las palabras reservadas son: and, as, assert, break, class, continue, def, del, elif, else, except, exec, finally, for, from, global, if, import, in, is, lambda, not, or, pass, print, raise, return, try, while, with, yield

## Cadenas

Las cadenas pueden ser definidas usando comilla simple (') o comilla doble ("). Una característica especial de las cadenas es que son inmutables, esto quiere decir que no se pueden cambiar los caracteres que contiene. El carácter \ sirve para escapar caracteres como \n o \t.

## Operadores

## Los operadores son muy similares a los de C:

* Aritméticos: +, -, \*, /
* Booleanos: and, not, or
* Comparación: >, <, >=, <=, ==

## Listas

* Son valores que están separados por comas dentro de paréntesis cuadrados.
* Está compuesta por cualquier cantidad y/o tipo de datos, ya sean cadenas, caracteres, números e inclusive otras listas.
* Se puede acceder a las listas por medio de índices, estos índices comienzan desde 0 hasta el número de elementos menos 1.
* Las listas son mutables.

## Tuplas

* Son parecidas a las listas, valores separados por una coma.
* Comparadas con las listas, las tuplas no son mutables.
* Se pueden aplicar las mismas operaciones que en las listas y su ventaja es que consumen menos memoria para almacenarse.
* Se crean, ya sea utilizando paréntesis o simplemente separando los valores por comas.

De esta hay un tipo especial, llamado tupla con nombre que, como su nombre lo indica, permite especificar un nombre para describirla.

## Diccionarios

* Un diccionario se crea usando { } y consta de dos partes: llave y valor.
* Las llaves son inmutables, deben de tener un solo tipo de dato, una cadena o número. Una vez que es creado, no se puede cambiar su tipo.
* Mientras que el valor puede ser de cualquier tipo y se puede cambiar con el tiempo.
* Los elementos en un diccionario no están ordenados.

## Funciones

* Una función o procedimiento sirve para empaquetar código que sirve para ser reutilizado.
* Se puede usar ese mismo código con diferentes entradas y obtener resultados o comportamiento de acuerdo con esos datos.

## Variables globales

El ambiente de ejecución es donde se efectúan las operaciones que componen un programa. Al momento de ejecutar un programa se crea un espacio de nombres para las variables. Hay dos tipos de espacio: global y local. Funcionan de la misma manera que lo hacen en C. Si la variable se declara fuera de las funciones pertenecen al espacio global,mientras que las variables que se definen dentro de una función pertenecen al espacio local, estas variables sólo pueden ser reconocidas y usadas dentro de la propia función.

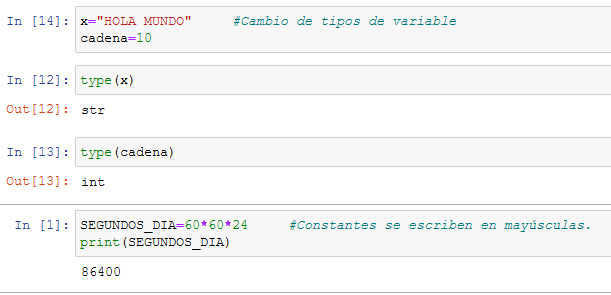
**Repositorio en Github de la práctica:**

[**https://github.com/RodrigoPG-W/Python-EDAI#python-edai**](https://github.com/RodrigoPG-W/Python-EDAI#python-edai)

# Desarrollo:

## Variables y tipos



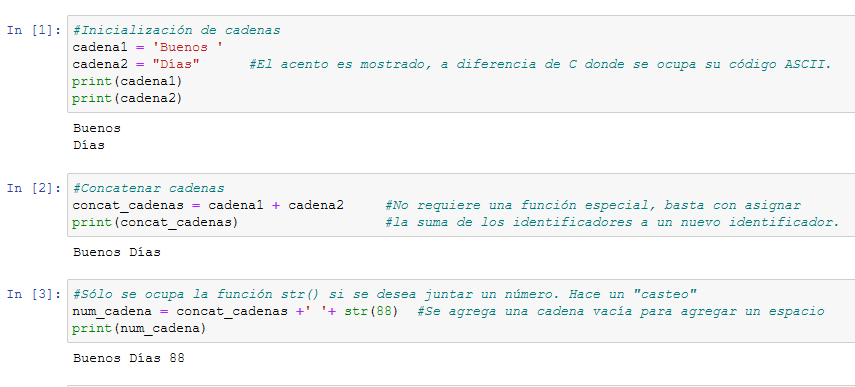


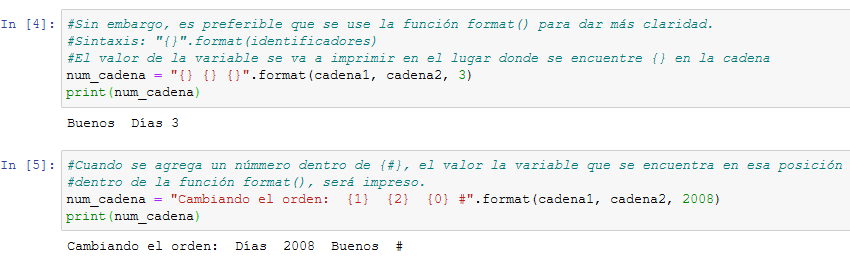
## 

## 

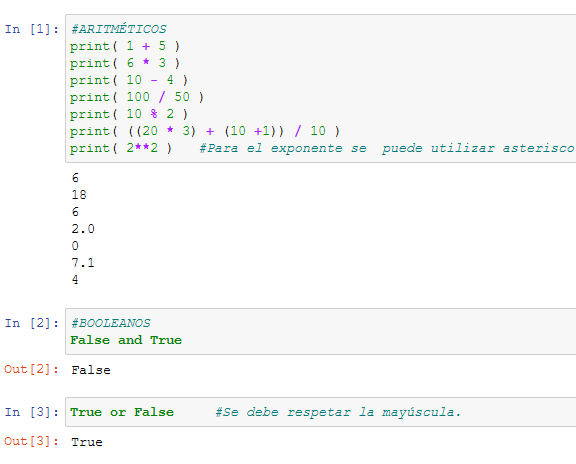
## 

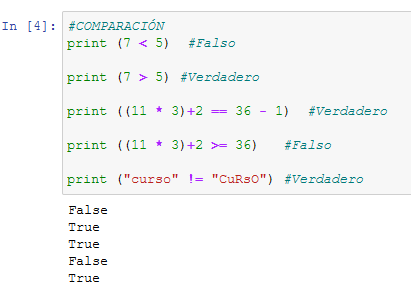
## Cadenas



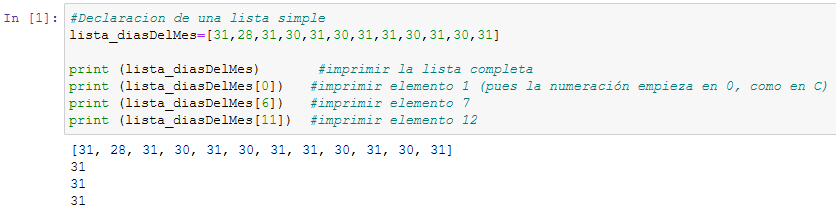


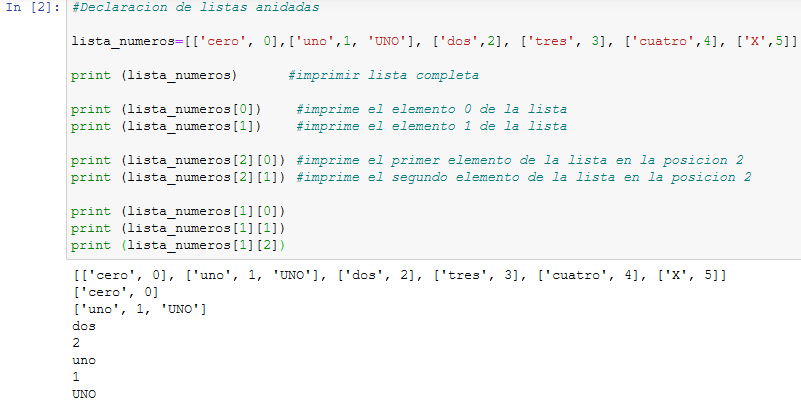
## Operadores

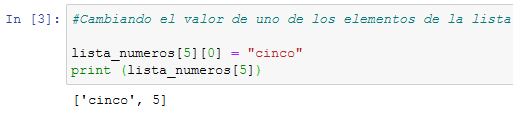




## Listas

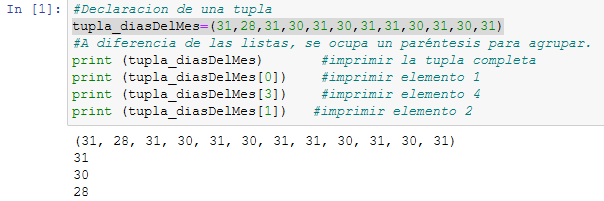


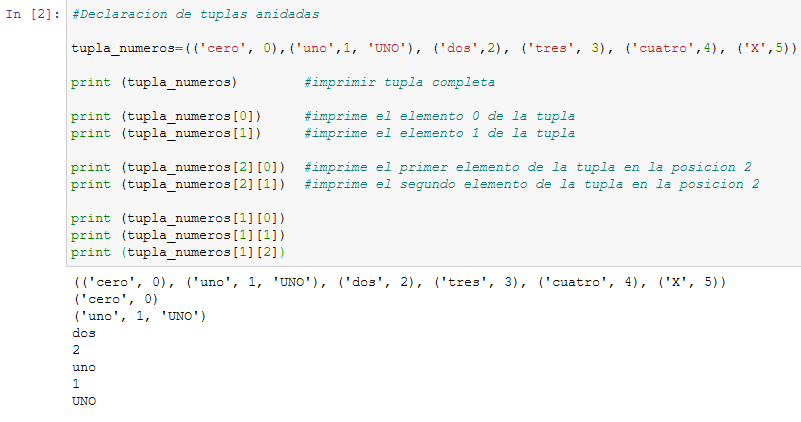


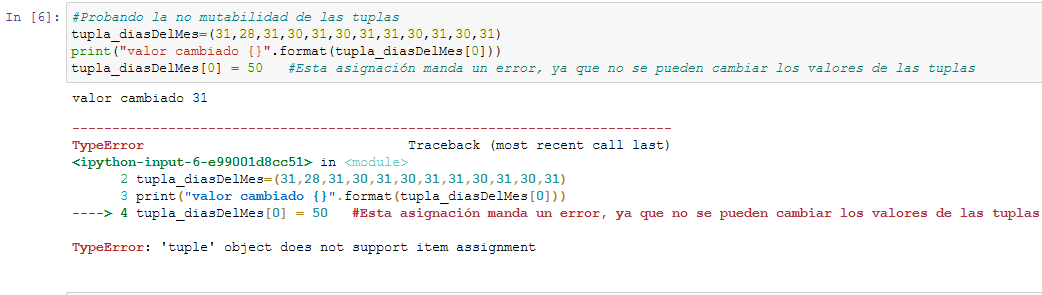


A pesar de poder ser “comparadas” con los arreglos en c. python es mucho más flexible y práctico al momento de asignar los nodos de datos en cada celda del arreglo.

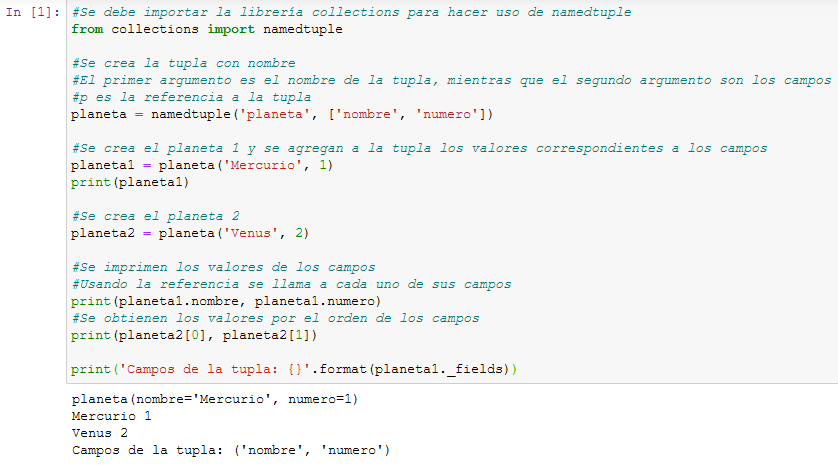
## Tuplas







## Tupla con nombre

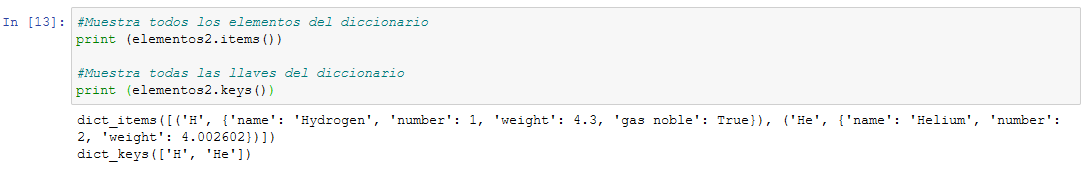


Algo a destacar es que se trabaja por medio de referencias, tal como las estructuras en lenguaje C. Tampoco es mutable, pues es una tupla.

## Diccionarios

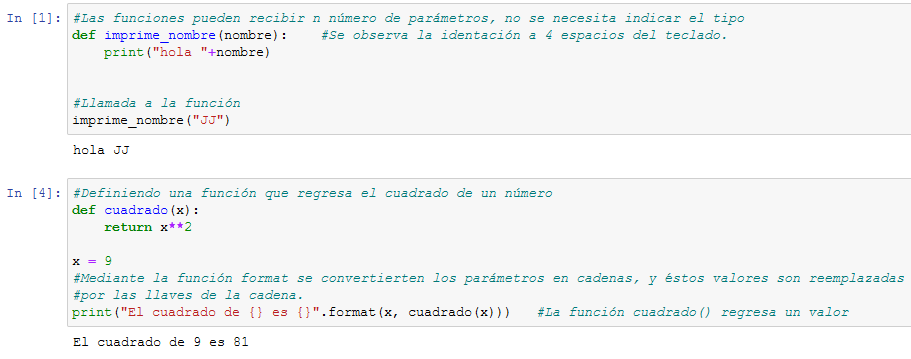


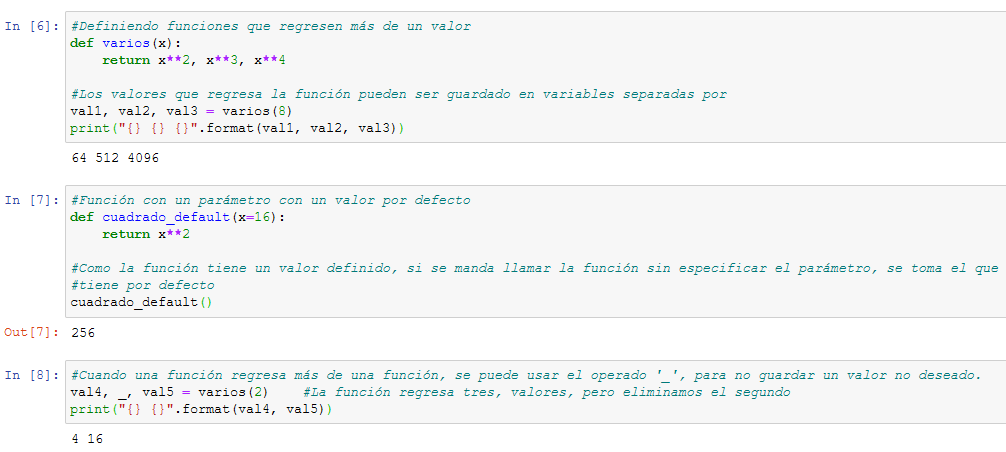




A pesar de que en la práctica se informa que los elementos pueden mostrarse desordenadas, tras manipular los valores y ejecutar varias veces, siempre se obtuvieron los resultados ordenados.

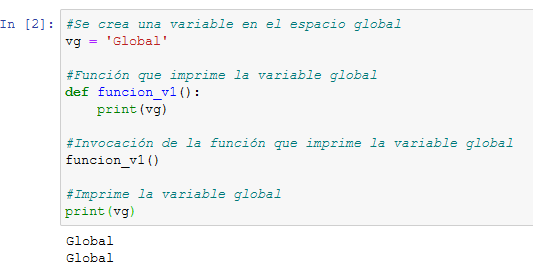
## Funciones

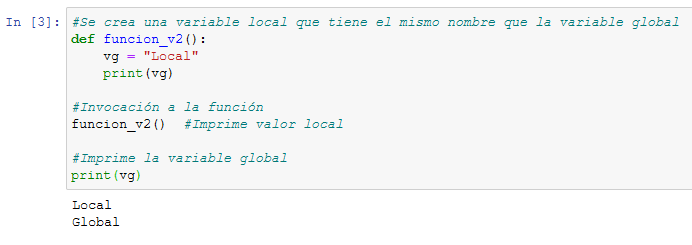


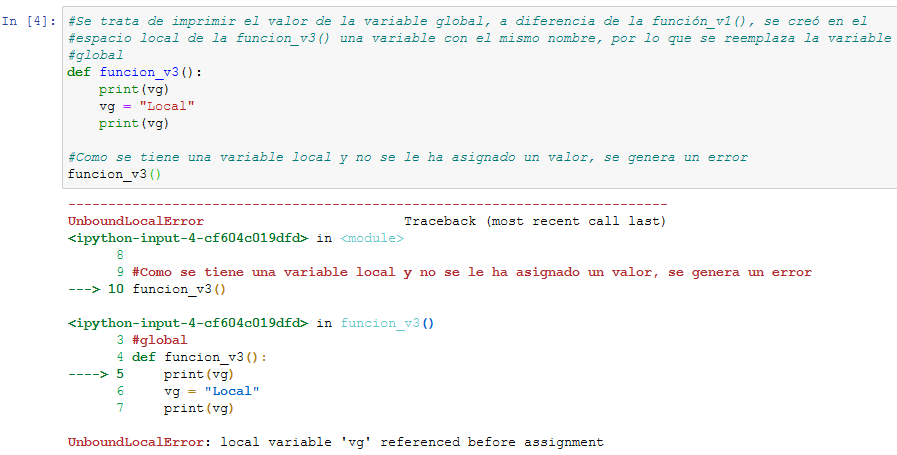


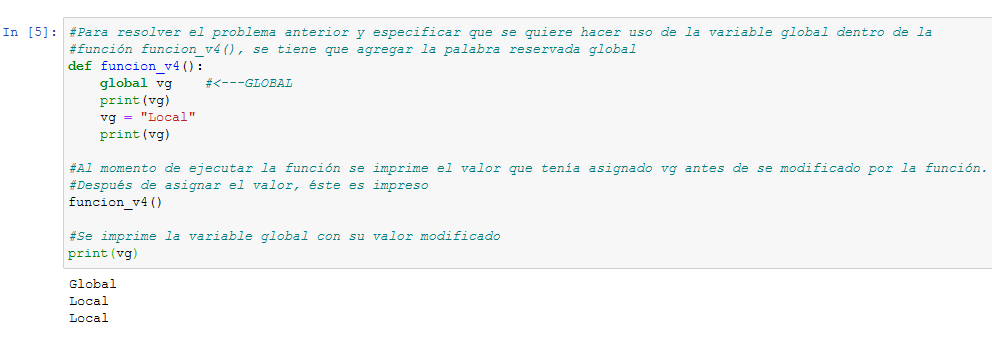
## Variables globales

El manejo de variables globales dentro de una función en el lenguaje Python se considera como una mala práctica, se recomienda que se pase como parámetro a la función y que se regrese un valor.









# Conclusión:

El lenguaje Python es una excelente herramienta para continuar en el aprendizaje como programador. 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, así como para empezar a generar un hábito de aprendizaje para seguir en constante actualización y seguirle el ritmo al desarrollo tecnológico del humano.

# Referencias:

* Python Software Foundation. *Tutorial de Python* (consultado el 31/03/2020). Recuperado de: <http://docs.python.org.ar/tutorial/3/real-index.html>
* Tutorial oficial de Python (consultado el 31/03/2020). Recuperado de: <https://docs.python.org/3/tutorial>
* Python Software Foundation. *Collections-High-performance container datatypes* (consultado el 31/03/2020). Recuperado de:<https://docs.python.org/2/library/collections.html>