**Objetivo:**

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

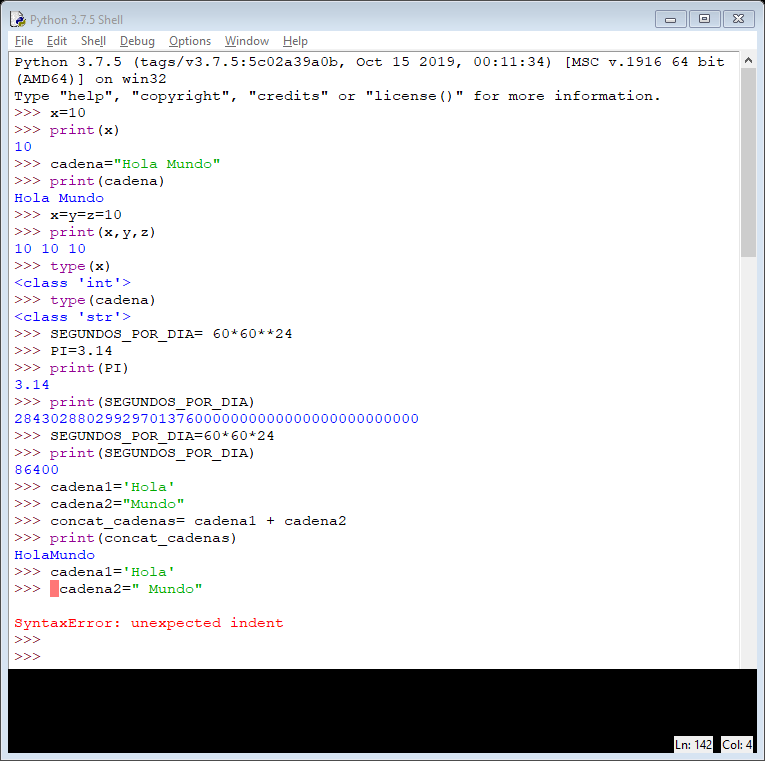
**Introducción.**

Variables y Tipos

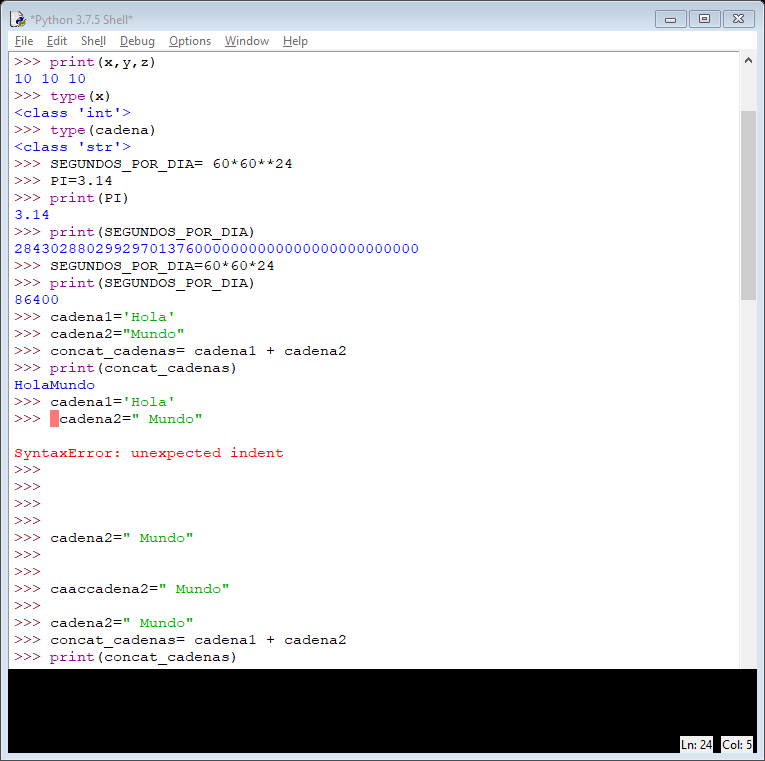
* Los nombres de las variables son alfanuméricos (a-z, A-Z, 0-9) y empiezan con una letra en minúscula.
* No se especifica el tipo de valor que una variable contiene, está implícito al momento de asignar un valor.
* No se necesita poner ; al final de cada instrucción.
* Mantener las indentaciones al momento de escribir código.

Nombres reservados en Python

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.

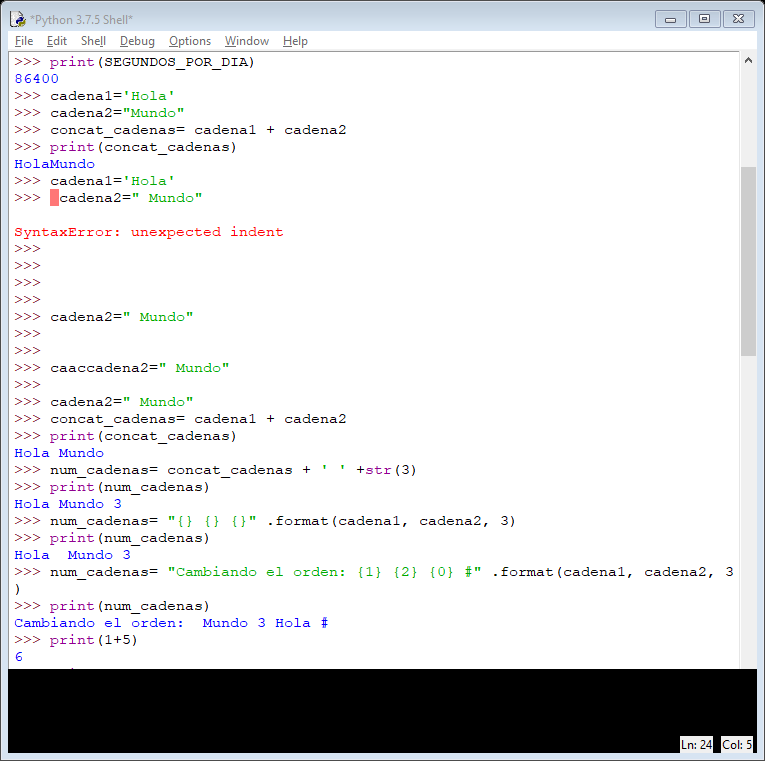


Cuando una variable tiene un valor constante, por convención, el nombre se escribe en mayúsculas.

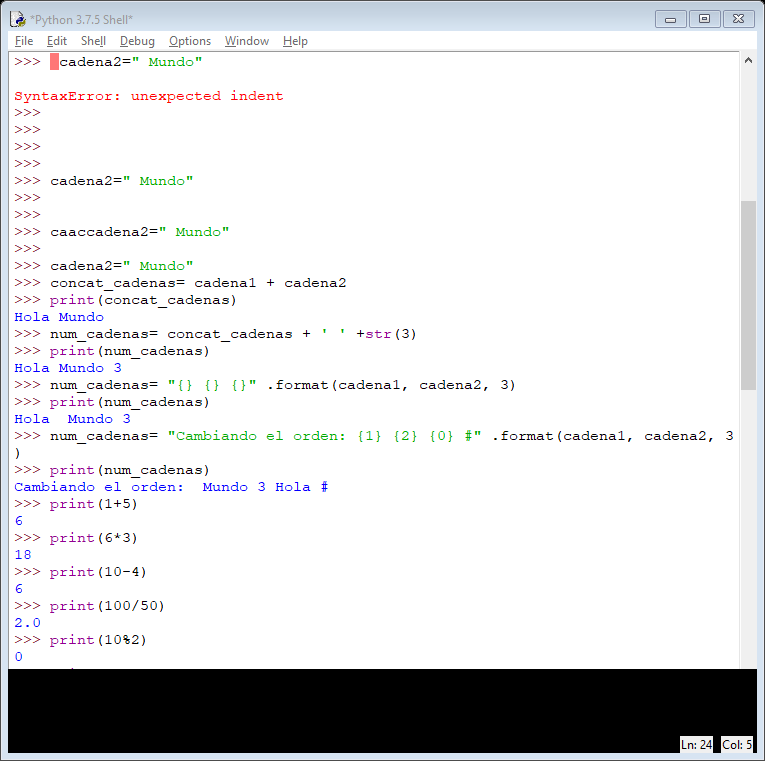


Cadenas

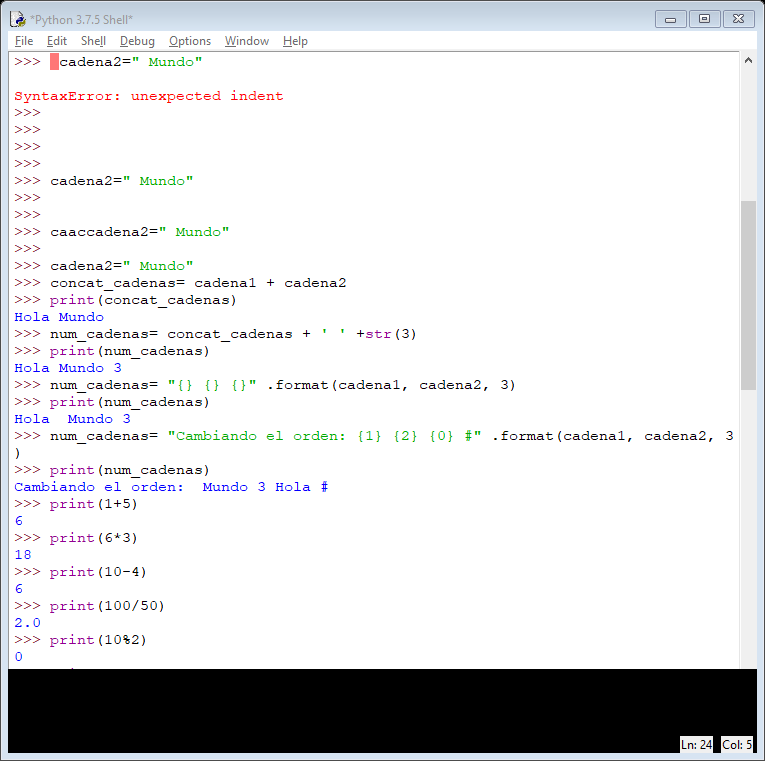
Las cadenas pueden ser definidas usando comilla simple (') o comilla doble ("). Una característica especial de las cadenas es que son inmutables, esto quiere decir que no se pueden cambiar los caracteres que contiene. El caracter \ sirve para escapar carcteres como \n o \t.



Para concatenar cadenas se recomienda el uso de la función format(), en lugar del viejo estilo del operador ‘+’.

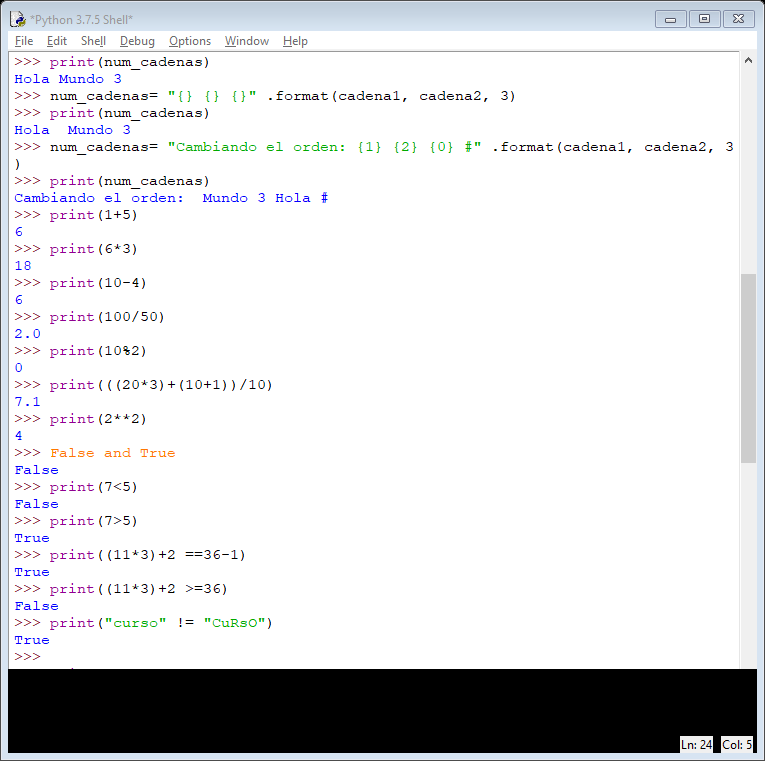


Por medio de la función format, se puede cambiar el orden en que se imprime las variables.

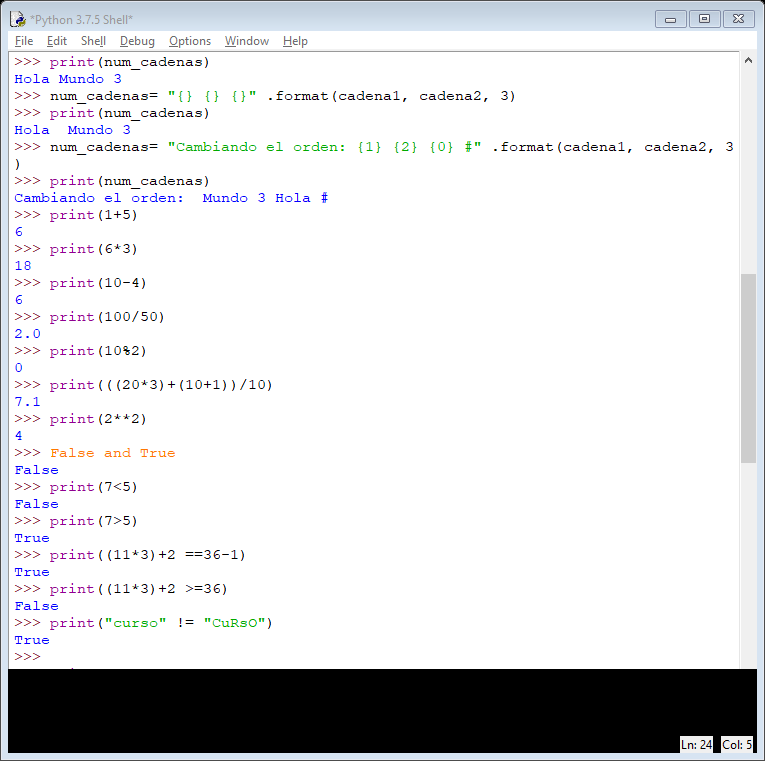


Operadores

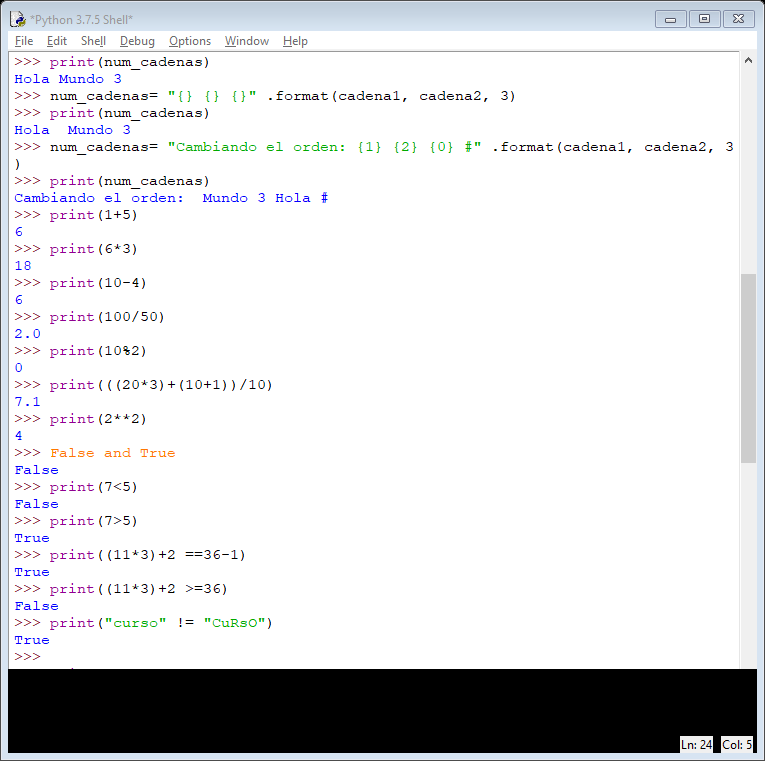
Aritméticos: +, -, \*, /



Booleanos: and, not, or

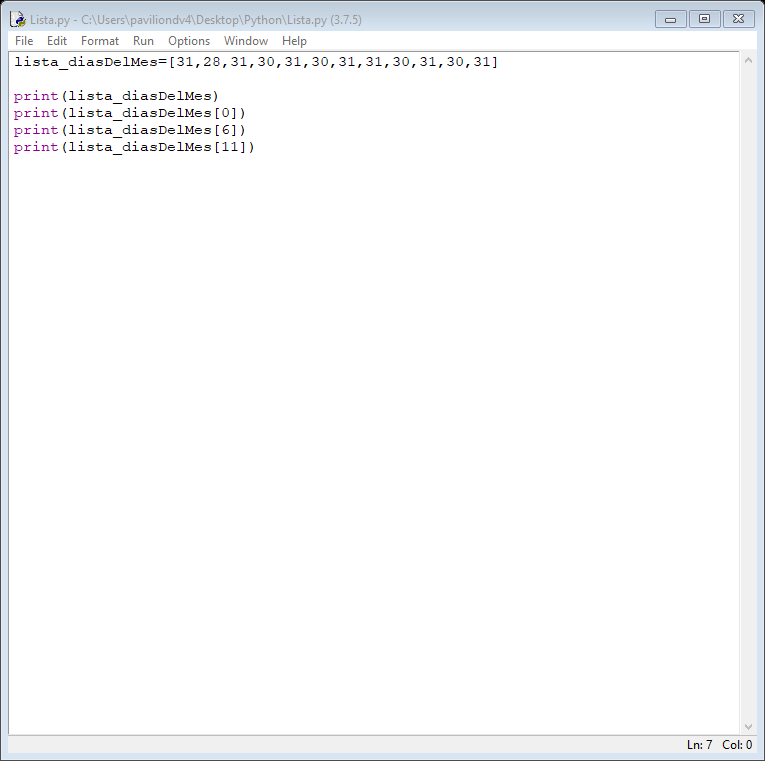


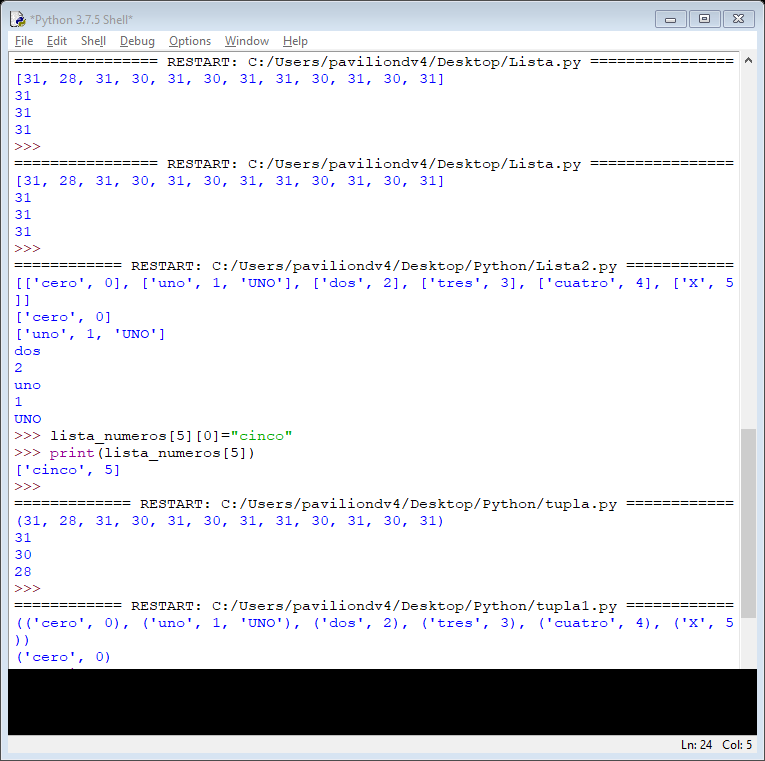
Comparación: >, <, >=, <=, ==

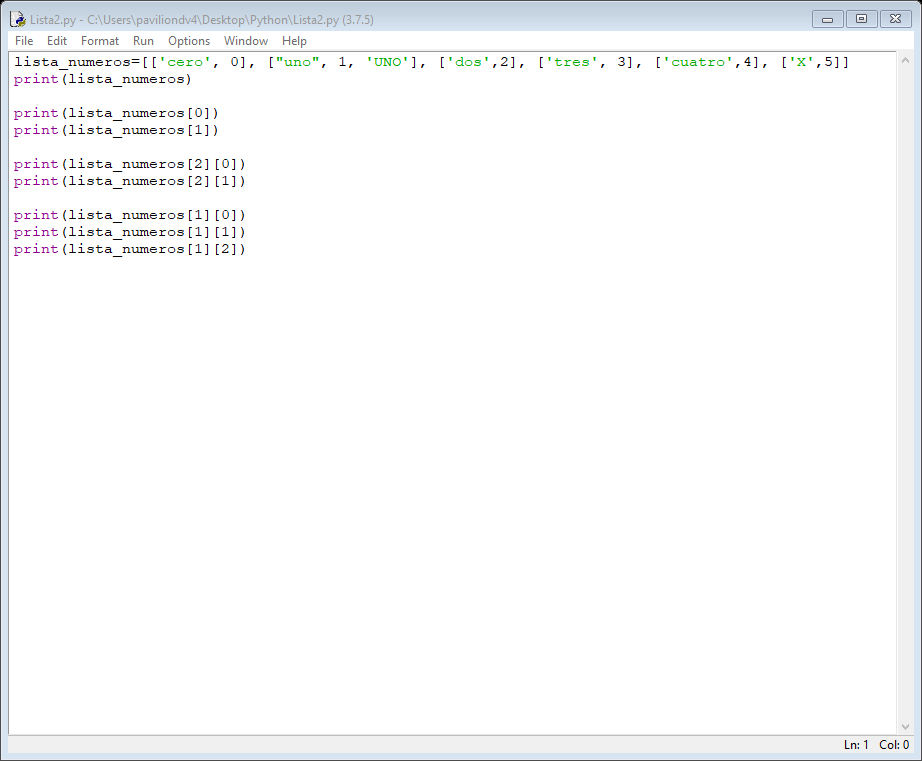


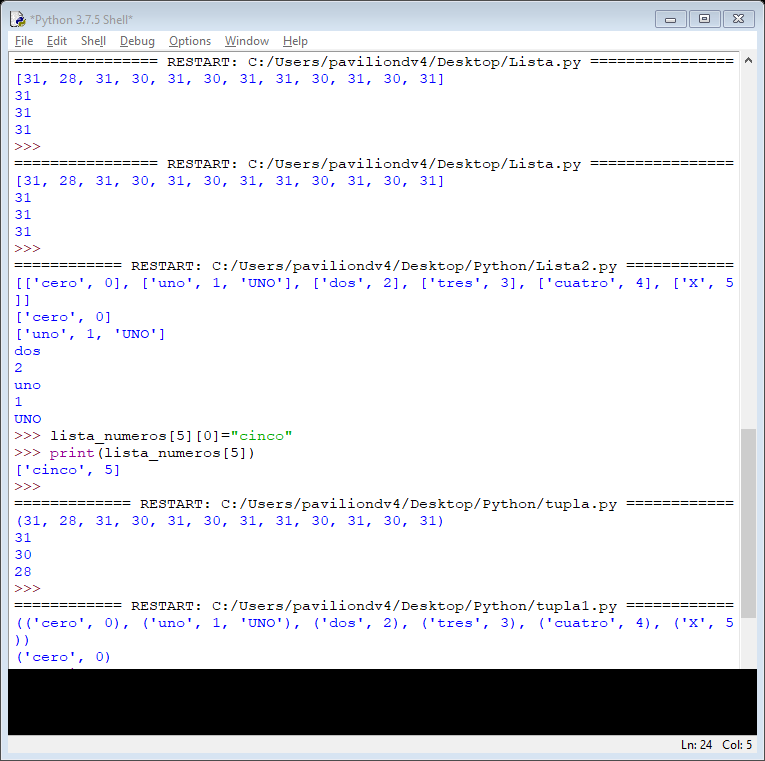
Listas

* Básicamente 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.

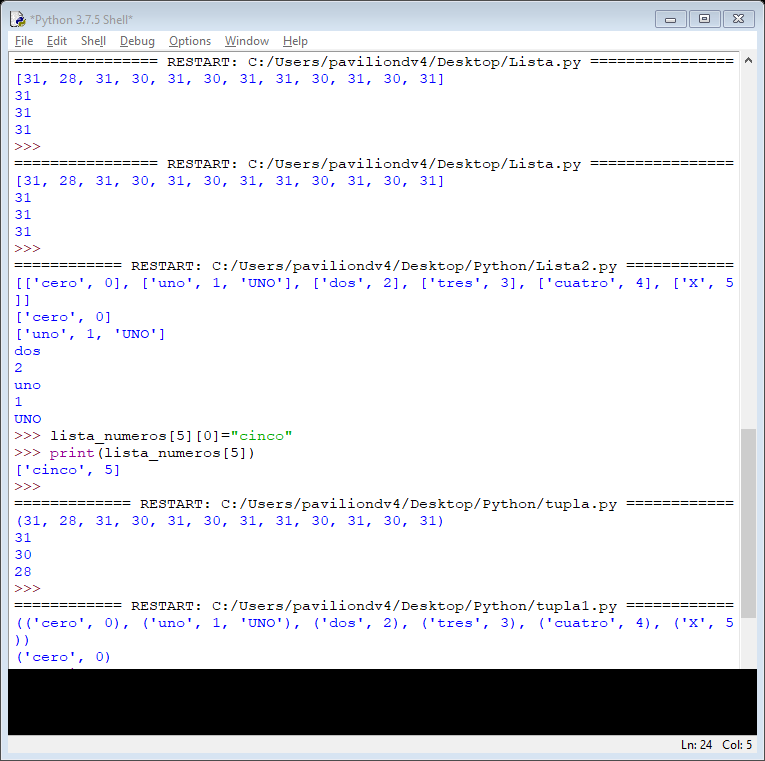






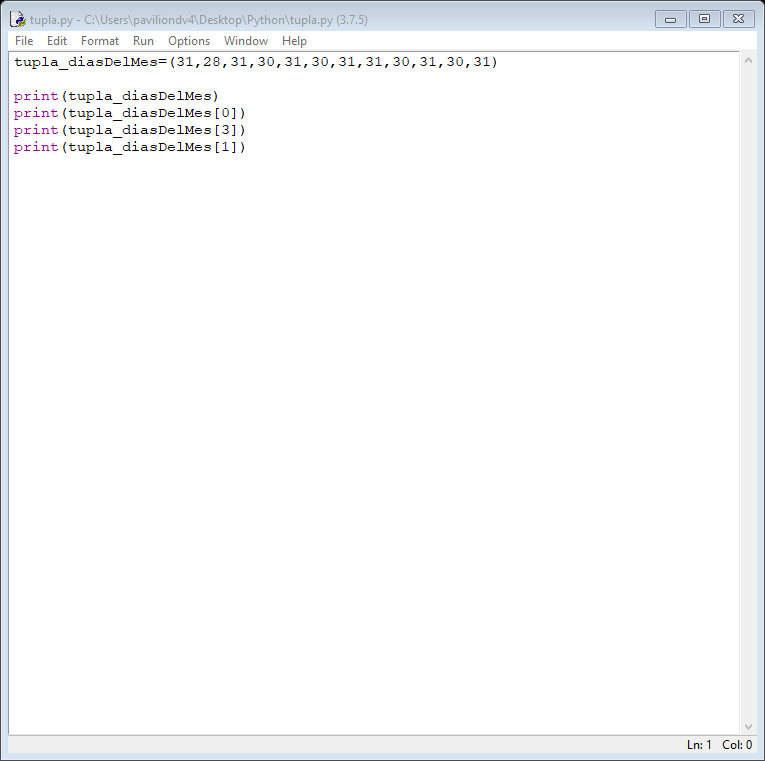


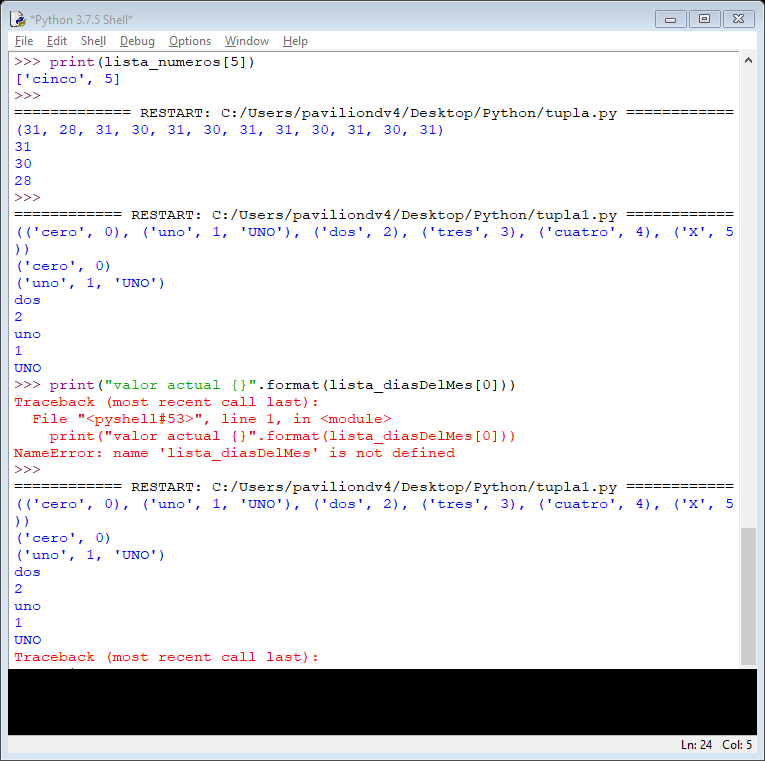
#Se cambia el valor de uno de los elementos de la lista

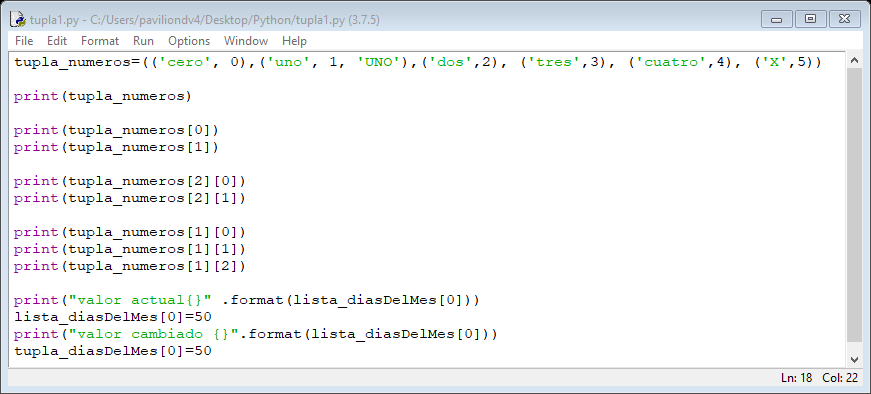


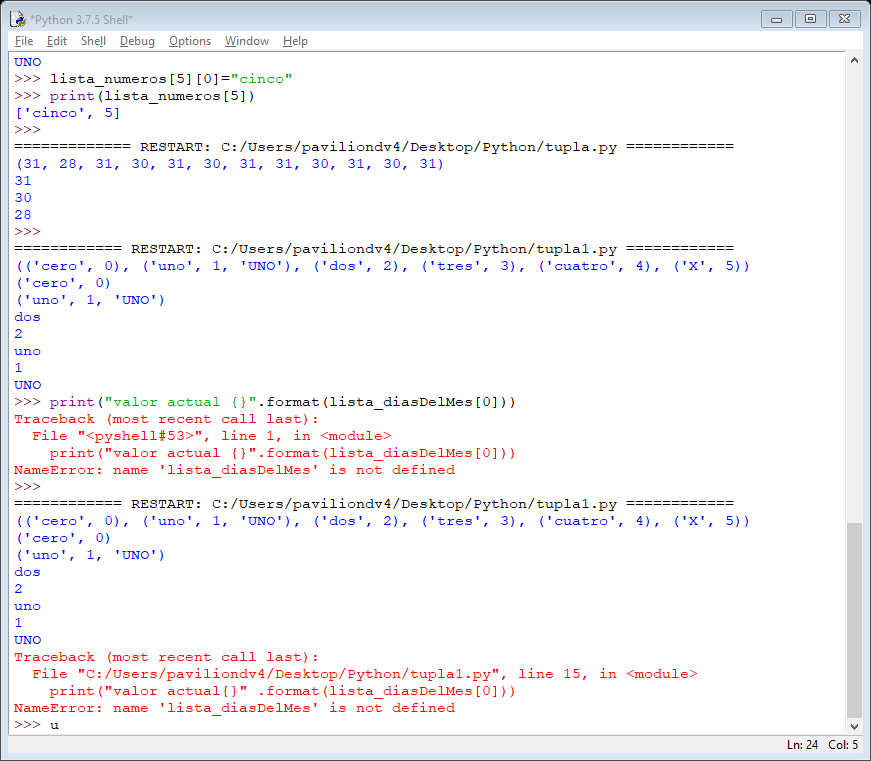
Tuplas

* 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.



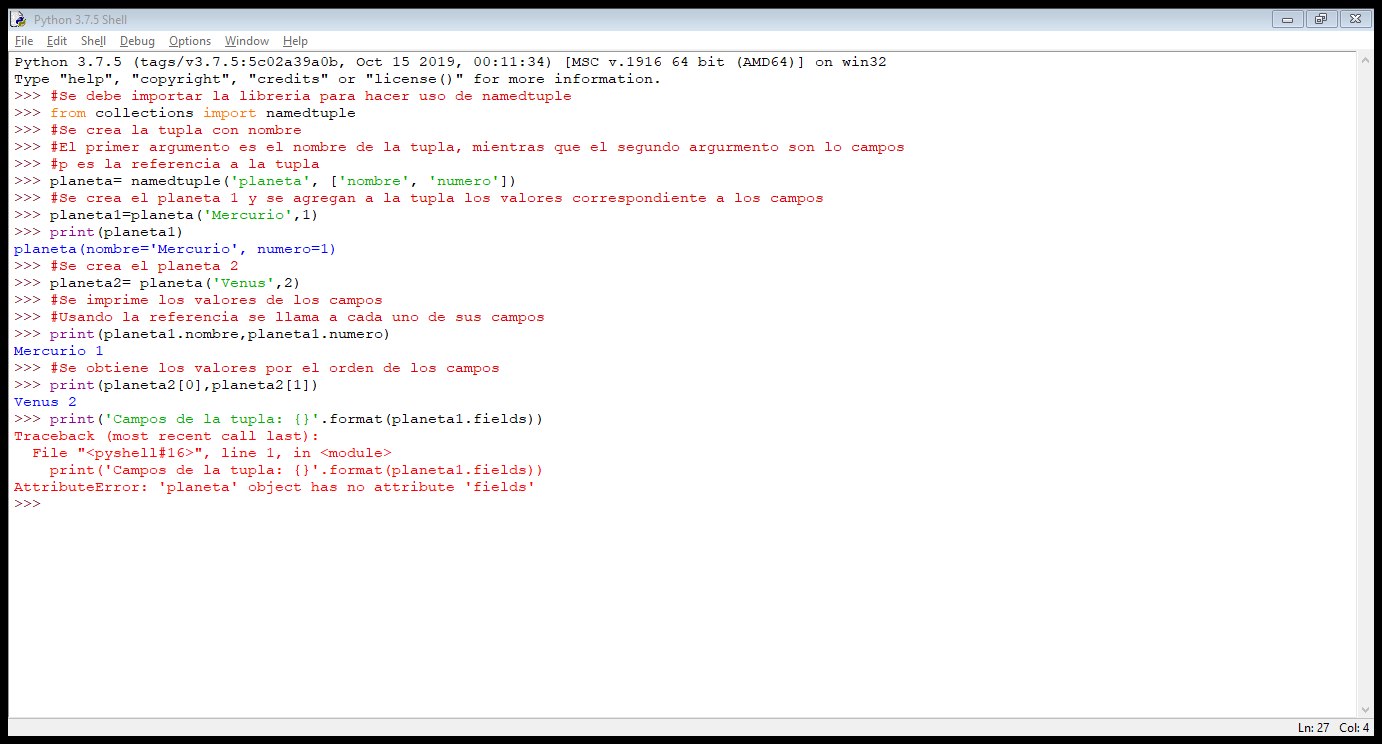






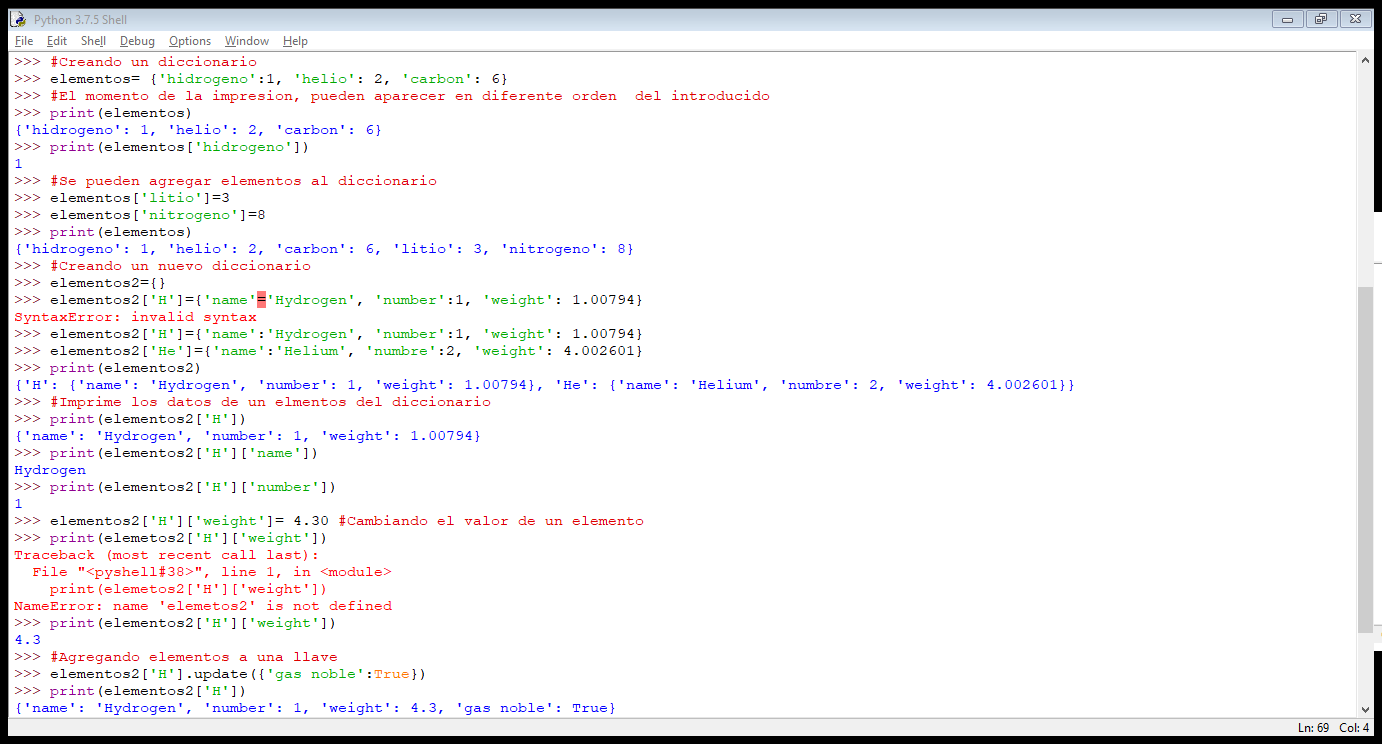
Tupla con nombre

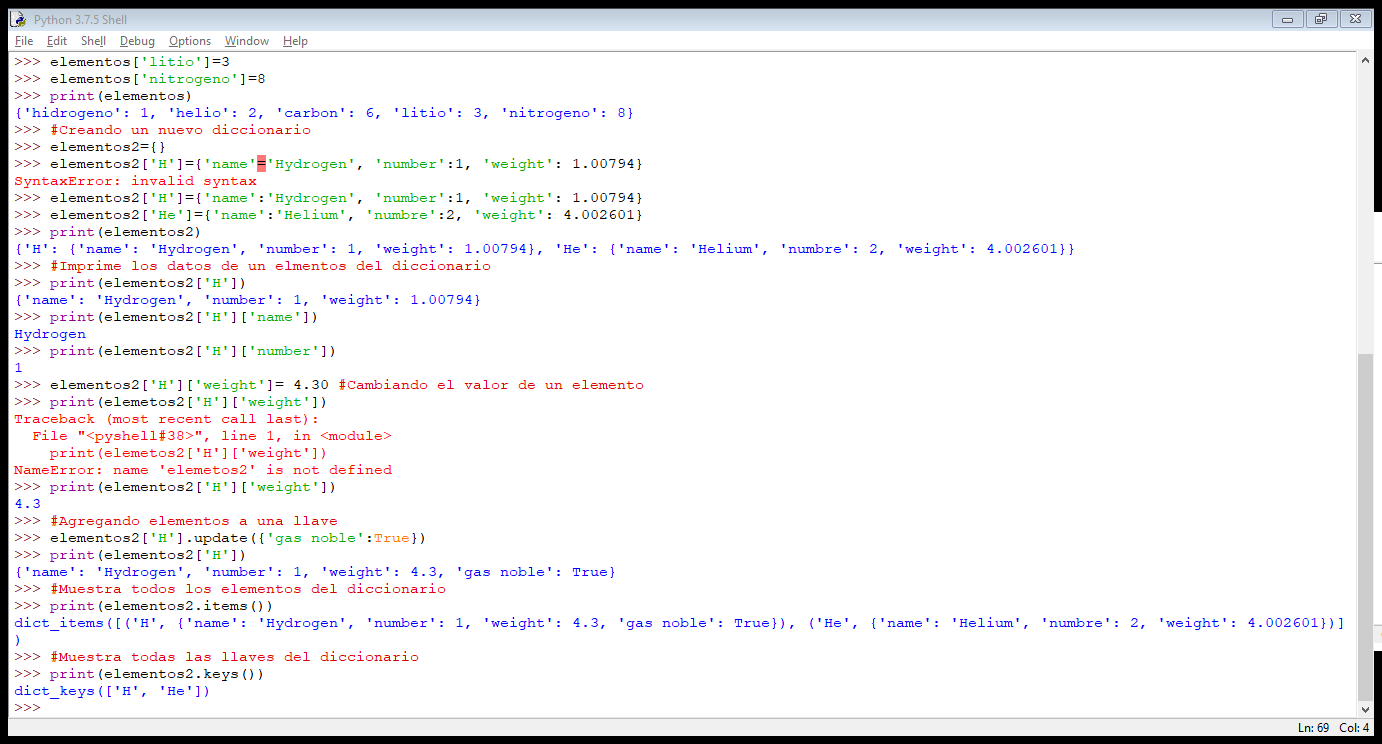
En comparación con la tupla anteriores, este tipo especial de tuplas permiten especificar un nombre para describirla.



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.





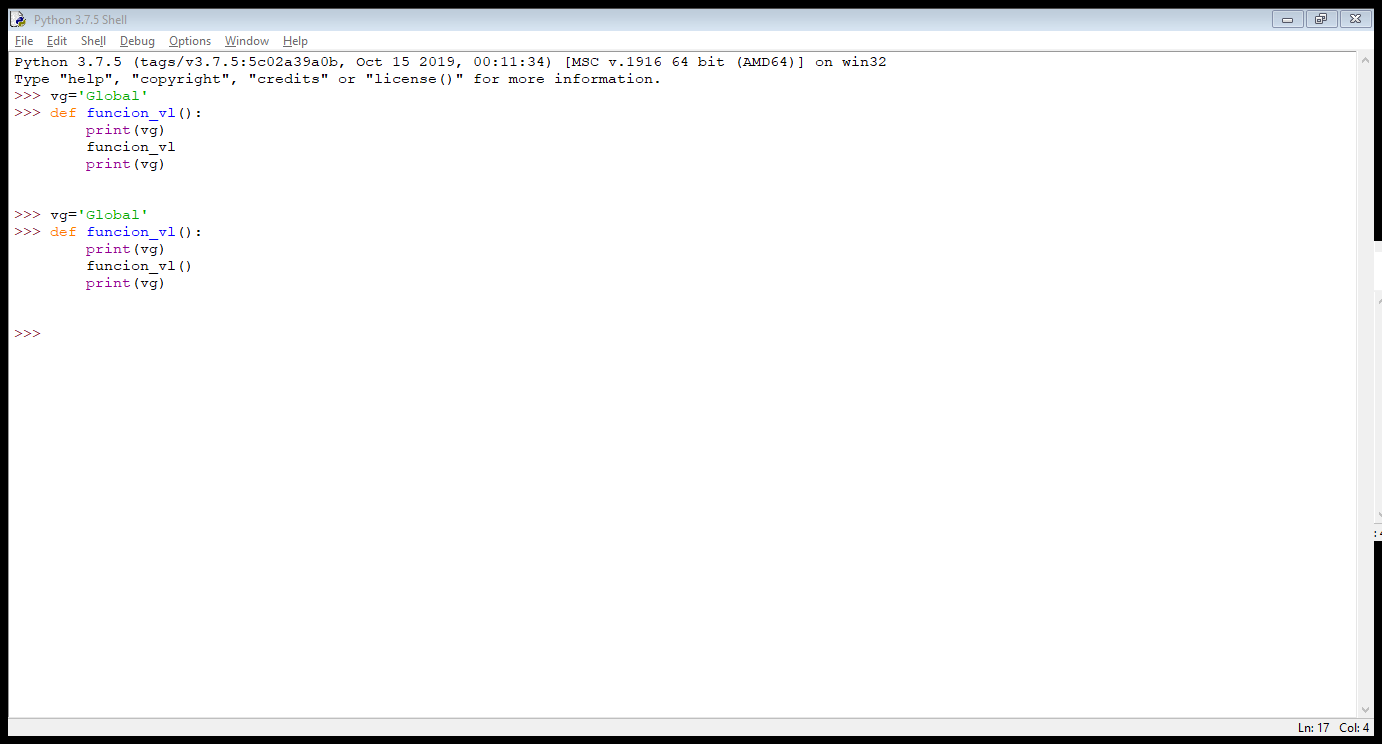
Funciones

* 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

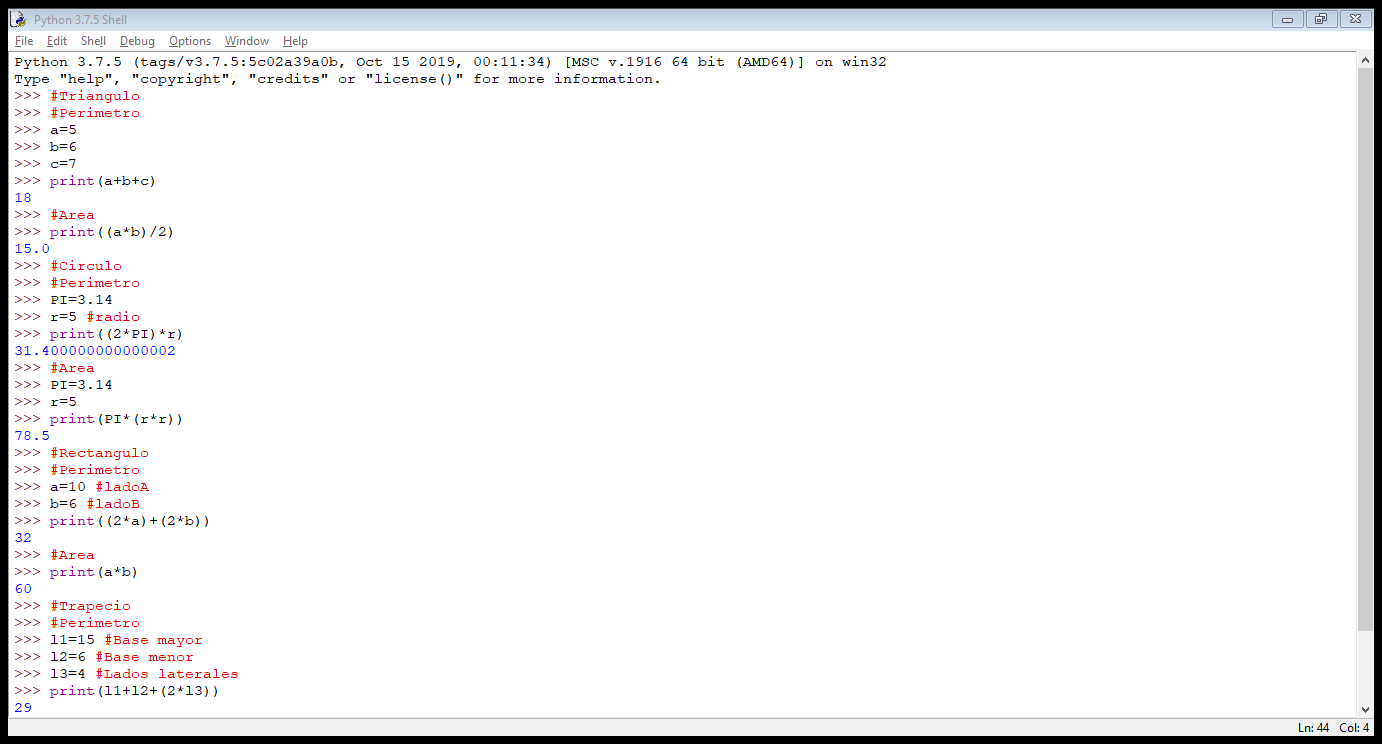
Se puede decir que el ambiente de ejecución es donde se efectúan las operaciones que componen un programa. Al momento de ejecutar un programa se crea un espacio de nombres para las variables. Hay dos tipos se espacio de nombres, el primero es el espacio global y el segundo el espacio local. Las variables que se declaren afuera de las funciones pertenecen al espacio global y no se necesita añadir un modificador para declararlas de esta manera. Por otro lado, todas 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.



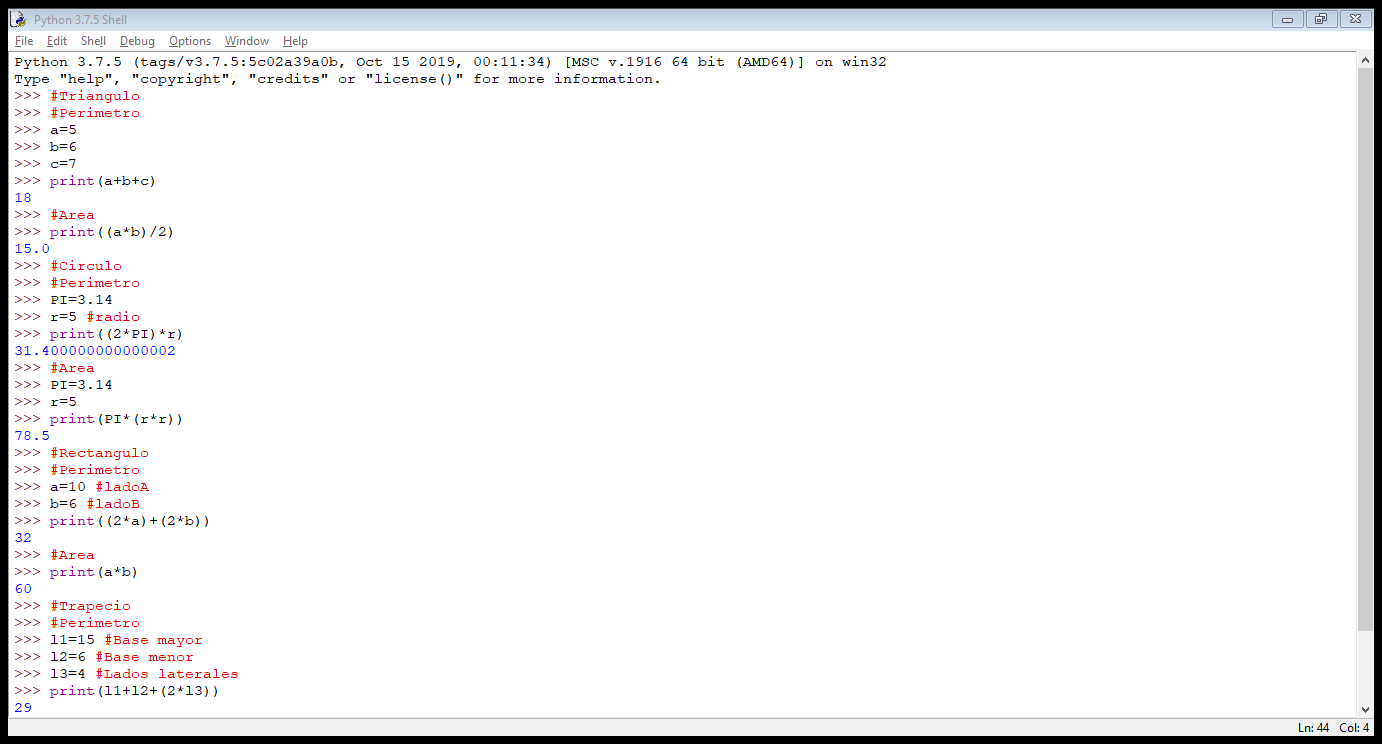
NOTA: El manejo de variables globales dentro de una función en el lenguaje Python se considera como una mala práctica, se recomienda que se pase como parámetro a la función y que se regrese un valor.

**Adicional**

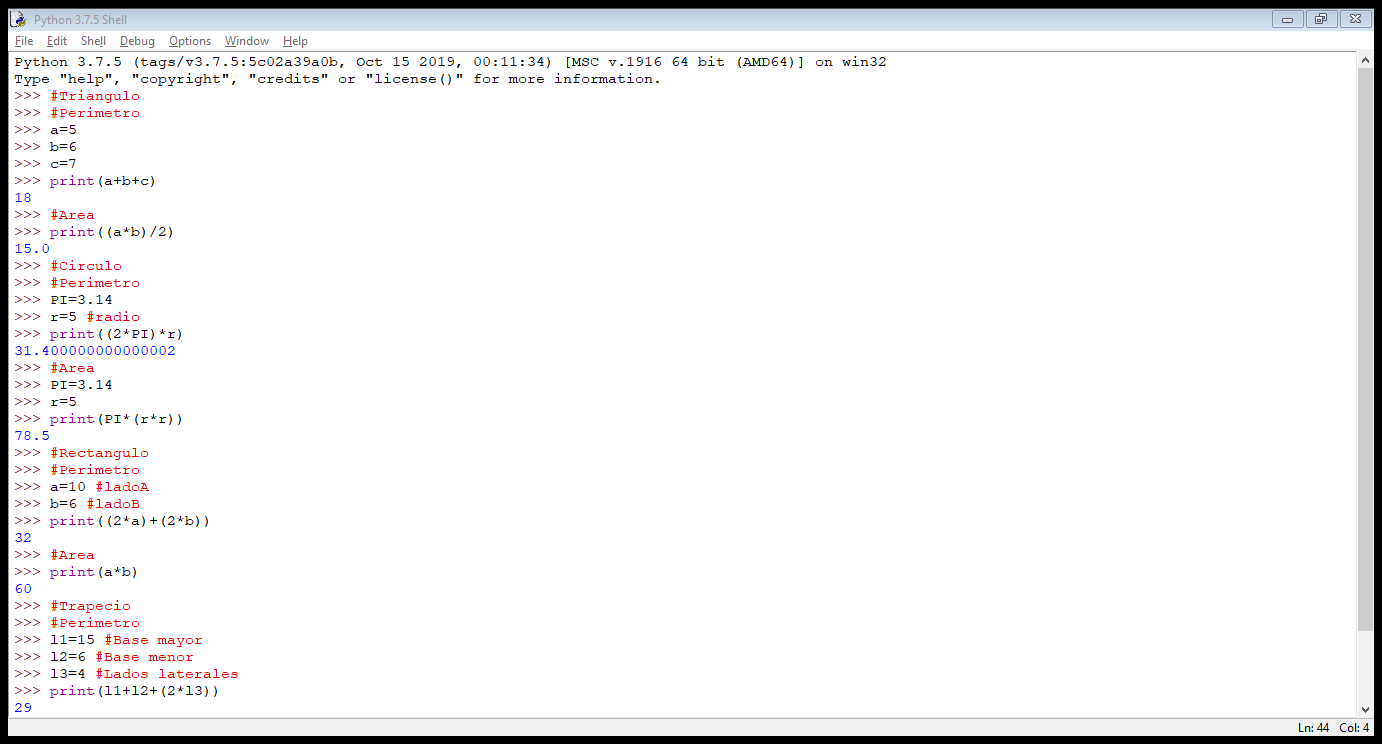
Área y perímetro de un triangulo



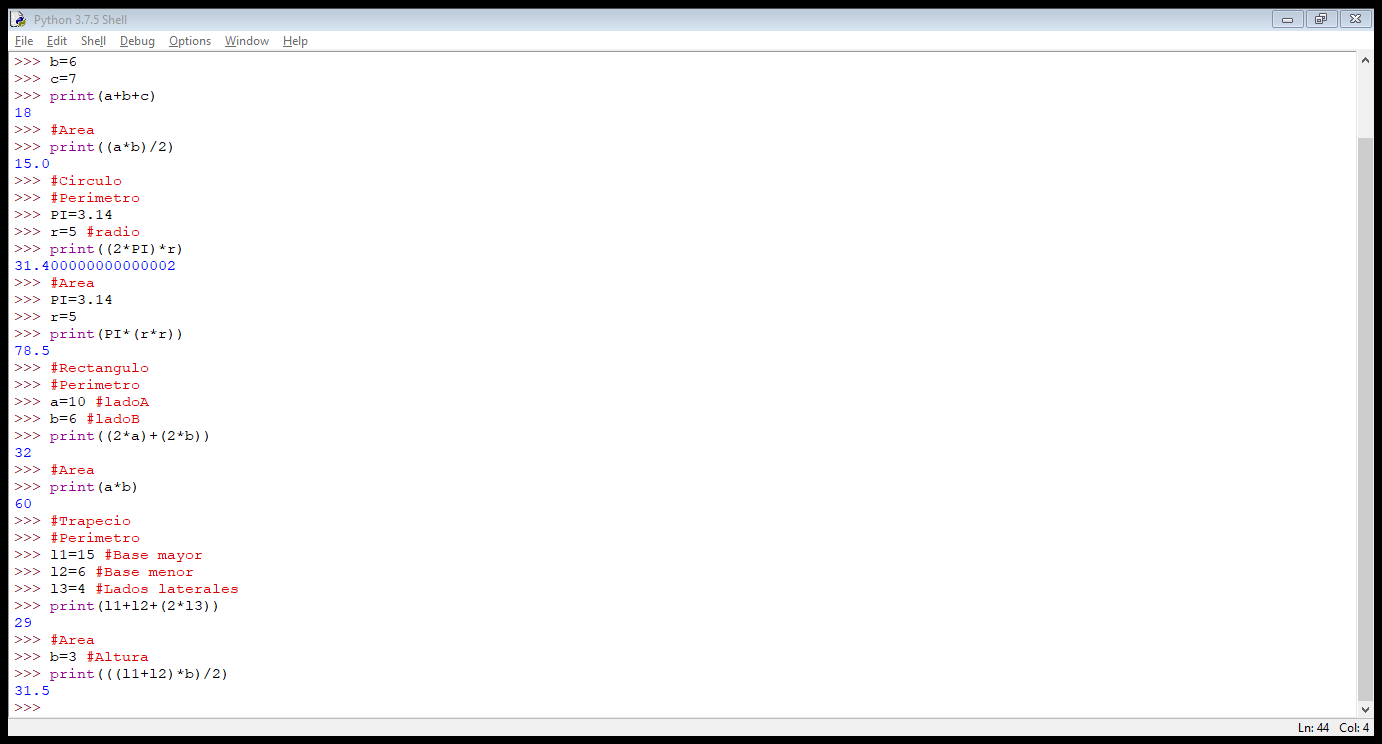
Área y perímetro de un círculo



Área y perímetro de un rectángulo



Área y perímetro de un trapecio



**Conclusión.**

Esta práctica me pareció muy interesante y muy divertida por una forma de aprender otro tipo de lenguaje de programación, aunque tuve algunos inconvenientes en la parte de funciones que no me daba, sin embargo los demás me pareció muy bien poder aprender más.

A mi parecer es un gran lenguaje de programación para trabajar y va influir en la construcción de nuevos proyectos para un futuro poder crear diversas aplicaciones. Espero aprender más y además poder conocer el IDE de Python.