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

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Marco Antonio Martinez Quintana |
| *Asignatura:* | Estructura de Datos y Algoritmos I |
| *Grupo:* | 17 |
| *No de Práctica(s):* | 9 |
| *Integrante(s):* | Abrego Abascal Diego |
| *No. de Equipo de cómputo empleado:* | - |
| *No. de Lista o Brigada:* | 1 |
| *Semestre:* | 2 |
| *Fecha de entrega:* | 31/03/2020 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**Introducción a Python I**

Introducción

“Python es un lenguaje de programación interpretado, orientado a objetos y de alto nivel con semántica dinámica. Sus estructuras de datos integradas de alto nivel, combinadas con tipeo dinámico y enlace dinámico, lo hacen muy atractivo para el desarrollo rápido de aplicaciones, así como para usarlo para scripting o pegamento para conectar componentes existentes. La sintaxis simple y fácil de aprender de Python enfatiza la legibilidad y, por lo tanto, reduce el costo del mantenimiento del programa. Python admite módulos y paquetes, lo que fomenta la modularidad del programa y la reutilización de código. El intérprete de Python y la extensa biblioteca estándar están disponibles en formato fuente o binario sin cargo para todas las plataformas principales, y se pueden distribuir libremente.

A menudo, los programadores se enamoran de Python debido a la mayor productividad que proporciona. Como no hay un paso de compilación, el ciclo de edición-prueba-depuración es increíblemente rápido. La depuración de programas de Python es fácil: un error o una entrada incorrecta nunca causará una falla de segmentación. En cambio, cuando el intérprete descubre un error, genera una excepción. Cuando el programa no detecta la excepción, el intérprete imprime un seguimiento de la pila. Un depurador de nivel fuente permite la inspección de variables locales y globales, la evaluación de expresiones arbitrarias, el establecimiento de breakpoints, el paso por el código línea por línea, etc. El depurador está escrito en Python, testificando el poder introspectivo de Python. Por otro lado, a menudo la forma más rápida de depurar un programa es agregar algunas declaraciones de impresión a la fuente: el ciclo rápido de edición, prueba y depuración hace que este enfoque simple sea muy efectivo.”

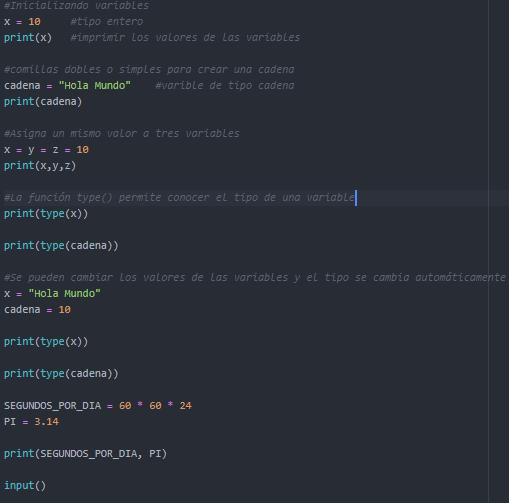
Traducción al español de: “*What is Python? Executive Summary*” de python.org

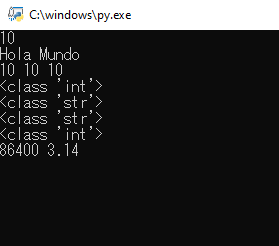
Objetivo

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

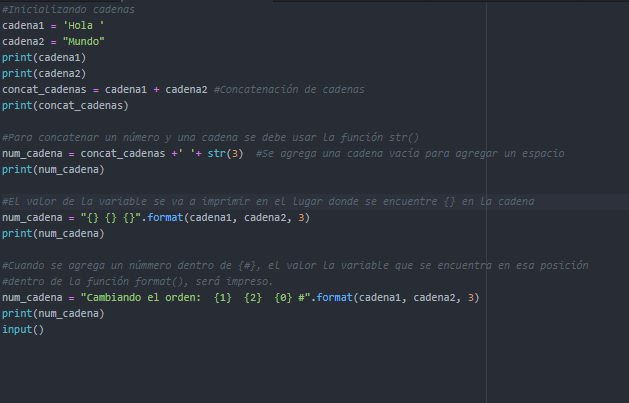
Desarrollo

1. Variables





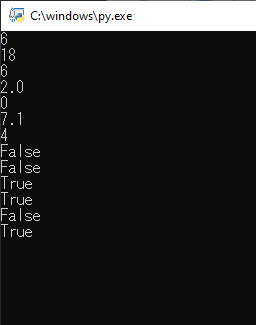
1. Cadenas



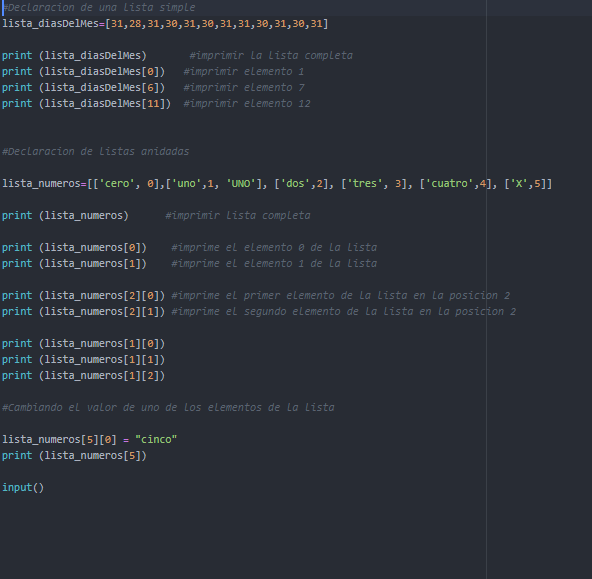


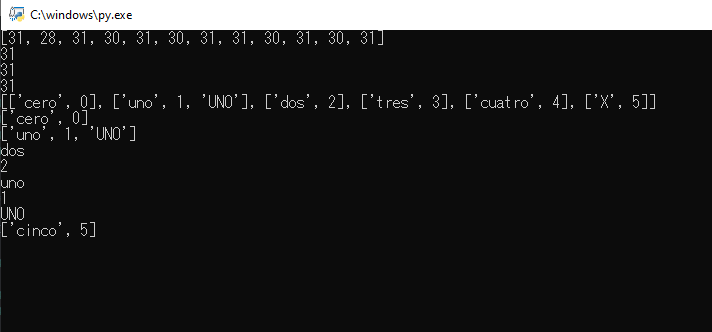
1. Operadores



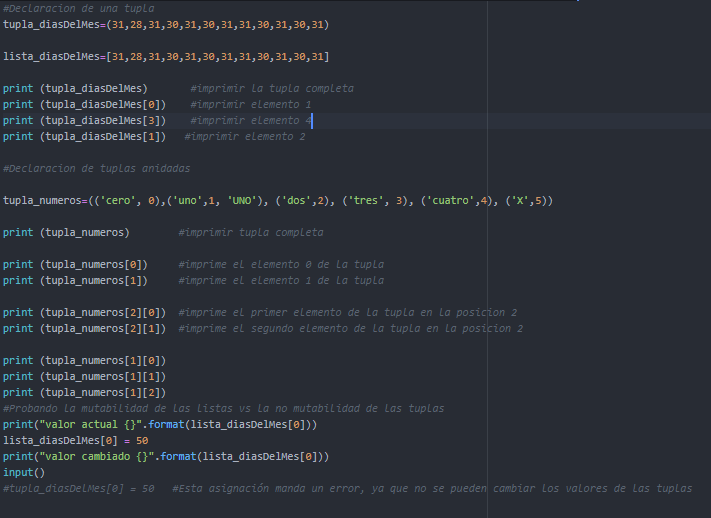


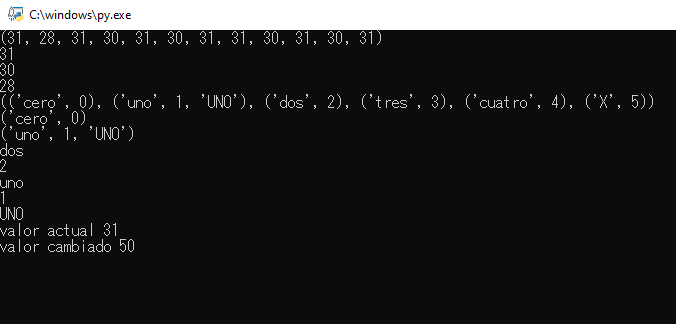
1. Listas



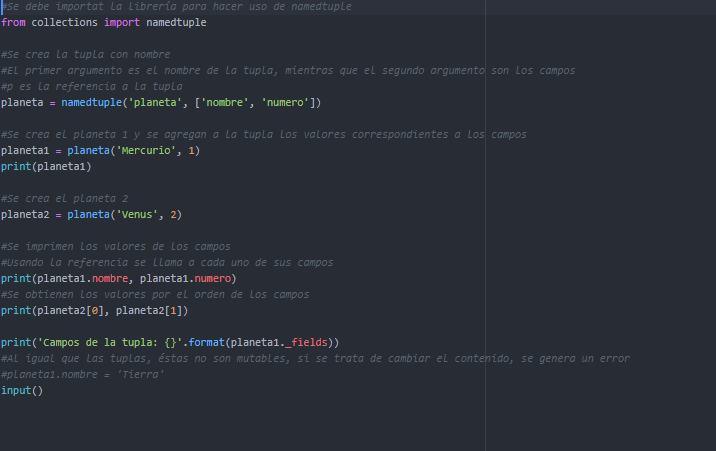


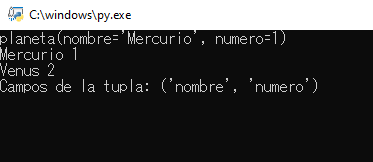
1. Tuplas



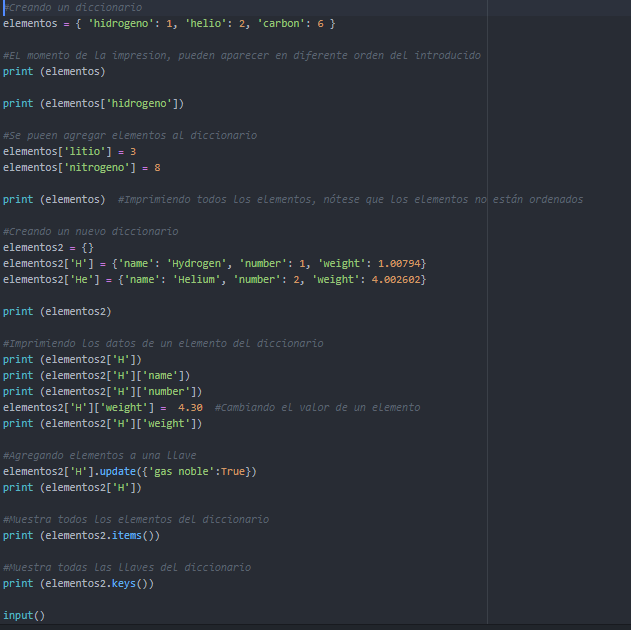


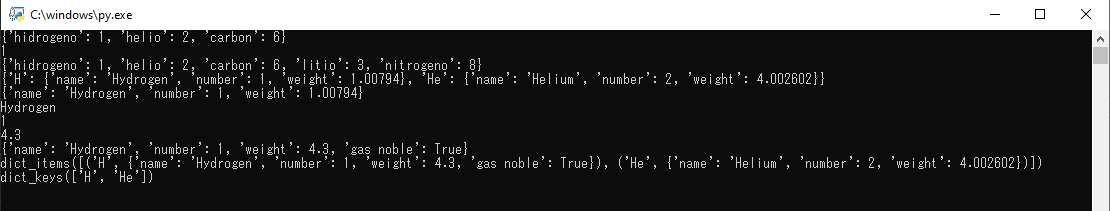
1. Tupla con nombre



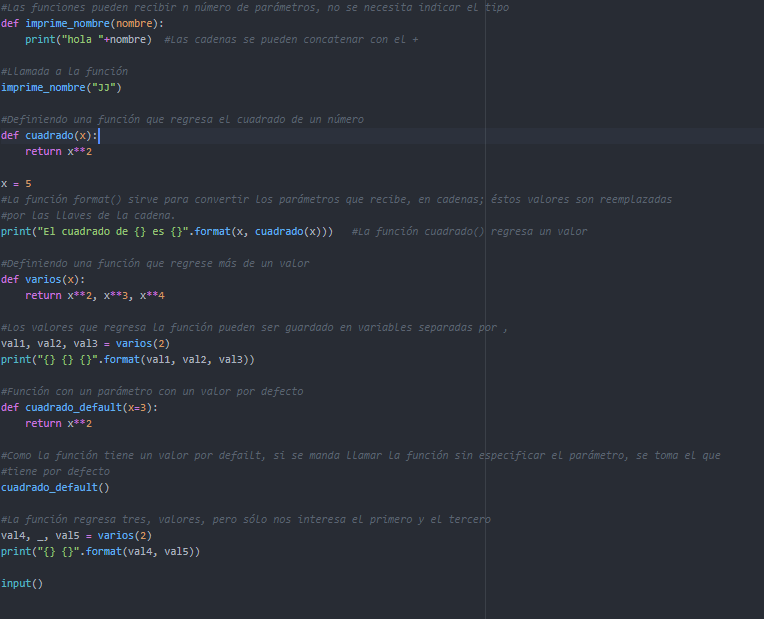


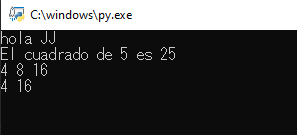
1. Diccionarios



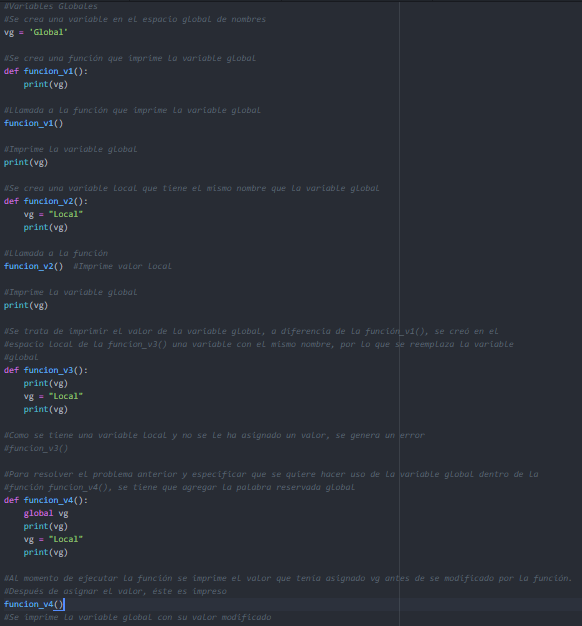


1. Funciones



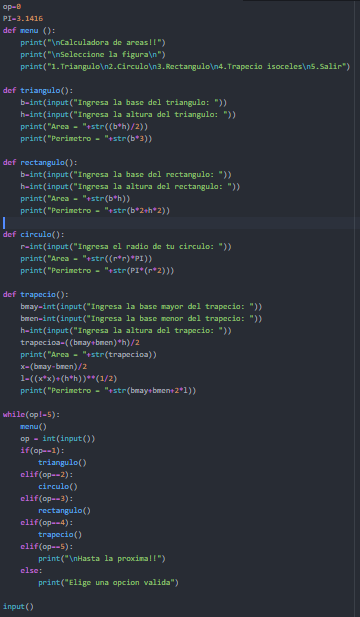


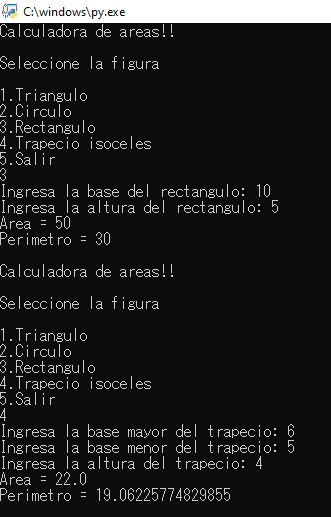
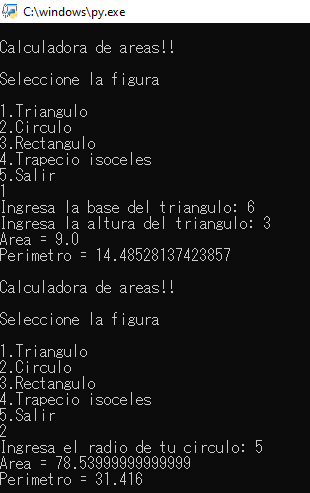
1. Funciones con variables Globales

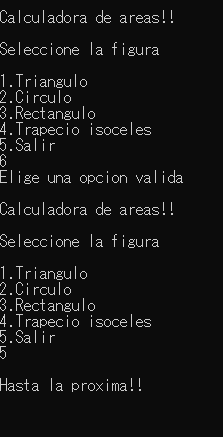




10. Calculadora de Áreas







Conclusión

En esta práctica se revisaron conceptos base del lenguaje de programación Python, como son: variables, manejo de cadenas de texto, operadores y funciones entre otros. Con esto se cimentaron las bases de manera practica del lenguaje de programación y a su vez se comprobó su sencillez y eficiencia a la hora de crear programas. Dicha eficiencia es proporcionada en gran medida por las características de alto nivel que proporciona el lenguaje, como su sintaxis simplificada que como consecuencia genera que la lectura y depuración del código se haga de manera rápida y sin complicaciones innecesarias por falta de orden y claridad. También es importante mencionar la gran importancia que representa que el lenguaje al ejecutarse con errores no regrese “*segmentation fault*” y en cambio regrese el seguimiento de la pila donde se presentó el error ya que esto como consecuencia genera que el debugging también se haga de manera mucho más rápida que en lenguajes que no cuentan con esta característica.

Al final de la practica se llevo a cabo una calculadora de áreas de formas geométricas donde se terminaron de acentuar los conocimientos adquiridos en esta práctica.

Concluyo que los objetivos de la practica se cumplieron satisfactoriamente ya que el ejercicio final se pudo realizar sin complicaciones.

Bibliografía

* Apuntes de clase
* http://lcp02.fi-b.unam.mx/static/docs/PRACTICAS\_EDA1/eda1\_p9.pdf
* https://www.python.org/doc/essays/blurb/