

Recuperatorio de introducción a la Programación en Python

Matemática para Economistas III

IDEI-UNGS.

25 de noviembre de 2022

*Sabes cuál es el problema?
Imaginate el algoritmo y no
programarlo*

Paul Huanca

Docente: Mateo Suster — msuster@campus.ungs.edu.ar —

Fecha límite de entrega: Martes 29/11/22 23:59hs

Pautas del examen

El examen deberá ser realizado a partir de los siguientes lineamientos:

1. Se debe resolver individualmente.
2. Se deberá entregar el código de Python utilizado en formato **.ipynb exclusivamente**.
3. **Nombrar** el archivo con el DNI del estudiante, seguido del texto *_recuperatorio1* (IMPORTANTE).
4. Se evaluará positivamente a la inclusión de las buenas prácticas de programación vistas en clase, así como la **prolijidad** y **originalidad** de las soluciones propuestas.
5. Es válido (y preferible) solicitar ayuda entre pares (alumnos y/o docente) por Slack antes que por otro medio. Su consulta no molesta (al contrario, enriquece).
6. Utilizar código ajeno no está prohibido, siempre y cuando se explicita la fuente correspondiente. Se podrá agregar todas las aclaraciones extras que se deseen o se consideren relevantes.
7. La entrega deberá realizarse a través de mail a la casilla msuster@campus.ungs.edu.ar con el asunto "MPE III - Recuperatorio 1 de Python".
8. Último, pero no menos importante: antes de entregar, revise que el código se ejecute de principio a fin sin errores !

Ejercicios del Recuperatorio

1. Definir un diccionario que contenga al menos tres (3) listas, cada una con una cotización de una moneda diferente (dolar, euro, yen, etc.) frente al peso, más una lista que indique el período temporal al que refiere cada una de las cotizaciones. Cada una de las listas debe tener al menos una longitud de cinco (5) momentos del tiempo distintos. A su vez, definir por fuera del diccionario, una lista con variables de tipo **float** que representen el valor de un inmueble valuado en pesos para los mismos momentos del tiempo. (Todos los números del ejercicio pueden ser reales o inventados).

A partir de dichos objetos, programar un bucle (`while` o `for`, aunque se recomienda este último por simplicidad) para calcular la evolución temporal de las cotizaciones del inmueble en cada una de las monedas seleccionadas. Guardar los resultados en un nuevo arreglo de datos (diccionario -recomendado por su orden- o lo que prefiera). Explicar el procedimiento.

Nota: prestar atención a cómo ordenar la estructura de control: hay varias formas de resolver el problema, por ejemplo, muy posiblemente se pueda llegar a necesitar un loop dentro de otro. Se recomienda también, por simplicidad, tratar de iterar sobre los diccionarios, aunque lo importante es arribar al resultado.

Nota 2: prestar atención a que el cálculo no se realice para el índice temporal, lo cual sería erróneo, pues es una lista más dentro del diccionario de cotizaciones (se recomienda utilizar una nueva estructura de control para evitarlo).

2. Hacer un gráfico a elección para visualizar el precio del inmueble para todas las cotizaciones, incluyendo en el eje x el momento del tiempo. **Sugerencia:** buscar en la documentación de la librería matplotlib un gráfico adecuado y citar la fuente.
3. Encapsular el programa del cálculo anterior en una función que devuelva tanto las distintas cotizaciones del inmueble y el gráfico **si y solo si todas las listas del diccionario de monedas** poseen el mismo largo. En caso contrario, la función debe devolver cuáles son las listas con mayor y menor observaciones. **Evaluar** la función con diferentes diccionarios de listas para revisar que se ejecute sin fallas y que los resultados sean los deseados. Denominar la función con un nombre representativo del algoritmo.
4. Construir una función propia que devuelva el menor número de una lista. Utilizar las estructuras de control vistas durante el curso (`if` y bucles `while` o `for`). Evaluar la función con distintas listas de números y también listas con strings (¿Por qué funciona también para estos casos?)