

Taller de Python

Guía de TPs

Introducción:

Esta guía de ejercicios está destinada a introducir a los alumnos en el mundo del lenguaje de programación más solicitado hoy en día. Comienza con programas de fácil resolución, con el objetivo de introducirse lentamente, y avanza hasta casos “reales” de uso cotidiano.

Los ejercicios se deberán resolver en formato digital, desarrollados en algún IDE, y utilizando los tutoriales y guías que consideren necesario.

Normativa de entrega de los TPs:

Cada grupo puede decidir entregar respetando alguno de estos lineamientos:

- Entrega de cada ejercicio por separado en formato script (cada uno en un archivo .py).
- Entrega de cada ejercicio por separado en un Colab Notebook.

La segunda forma de entrega es la recomendada por el equipo docente.

Tutoriales de Python:

A continuación se proveen links de tutoriales oficiales de Python, junto con links y recomendaciones varias.

Tutoriales oficiales:

- [Getting started in Python](#)
- [Python Beginner's Guide](#)
- [Python Developer's Guide](#)
- [Tutorial de Python](#) (basado en Python 3.10.0)

Links varios:

- [Página oficial](#)
- [Intérprete online](#)
- Visual Studio Code: mejor editor de código.
- Sublime Text: otro excelente editor de código.
- [Python cheatsheet collection](#)
- [Google Colab notebooks](#)

Ejercicios obligatorios de Python

1) Manejo de strings

Escribir un programa que pregunte el nombre completo del usuario en la consola y después muestre por pantalla el nombre completo del usuario tres veces, una con todas las letras minúsculas, otra con todas las letras mayúsculas y otra solo con la primera letra del nombre y de los apellidos en mayúscula. El usuario puede introducir su nombre combinando mayúsculas y minúsculas como quiera.

2) Cálculos elementales

Escribir un programa que pida al usuario dos números enteros y muestre por pantalla: *<n> entre <m> da un cociente <c> y un resto <r>*, donde *<n>* y *<m>* son los números introducidos por el usuario, y *<c>* y *<r>* son el cociente y el resto de la división entera respectivamente.

3) Uso de diccionarios

Escribir un programa que guarde en una variable el diccionario `{'Peso': '$', 'Dolar': 'U$', 'Yen': '¥'}`, pregunte al usuario por una divisa y muestre su símbolo o un mensaje de aviso si la divisa no está en el diccionario.

4) Depuración

Corregir los errores del siguiente programa:

```
contraseña = input('Introduce la contraseña: ")
if contraseña in ['sesamo']:
    print('Pasa')
else
    print('No pasa')
```

5) Creación de clases

Escribir una clase **Caja** para representar cuánto dinero hay en una caja de un negocio, desglosado por tipo de billete (por ejemplo, en el kiosco de la esquina hay 6 billetes de 500 pesos, 7 de 100 pesos y 4 monedas de 2 pesos). Las denominaciones permitidas son 1, 2, 5, 10, 20, 50, 100, 200, 500 y 1000 pesos. Debe comportarse según el siguiente ejemplo:

```

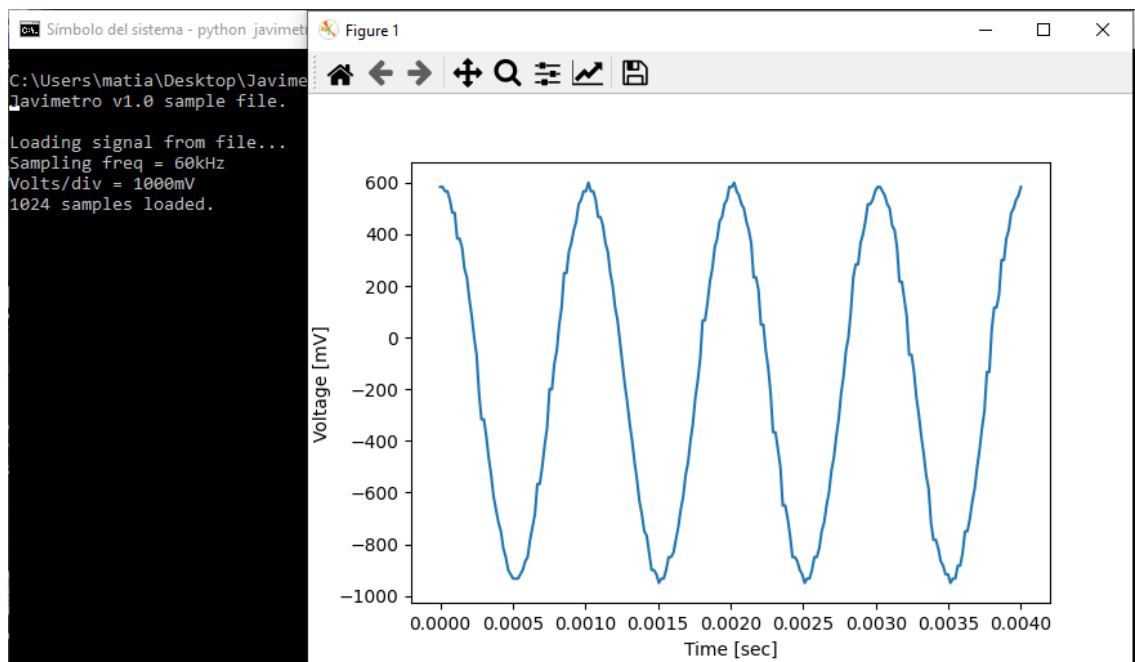
>>> c = Caja({500: 6, 300: 7, 2: 4})
ValueError: Denominación "300" no permitida
>>> c = Caja({500: 6, 100: 7, 2: 4})
>>> str(c)
'Caja {500: 6, 100: 7, 2: 4} total: 3708 pesos'
>>> c.agregar({250: 2})
ValueError: Denominación "250" no permitida
>>> c.agregar({50: 2, 2: 1})
>>> str(c)
'Caja {500: 6, 100: 7, 50: 2, 2: 5} total: 3810 pesos'

>>> c.quitar({50: 3, 100: 1})
ValueError: No hay suficientes billetes de denominación "50"
>>> c.quitar({50: 2, 100: 1})
200
>>> str(c)
'Caja {500: 6, 100: 6, 2: 5} total: 3610 pesos'
  
```

5) Graficado de señal

En el laboratorio de electrónica se tomaron mediciones con un osciloscopio de un filtro pasa bajos. Dicho osciloscopio permite guardar una captura de la señal en un pendrive, arrojando el archivo *SAMPLES.TXT*.

Se pide escribir un programa que lea dicho archivo y grafique un intervalo representativo de la señal guardada. Se recomiendan 240 muestras. El resultado final responde a:



Nota: El archivo posee un encabezado con datos de la medición, y luego los valores medidos en un formato separado por comas.

Sugerencia: Puede utilizar la librería *matplotlib* para graficar la señal.

Sugerencia: Vaya encarando el ejercicio de a poco, dividiéndolo en partes más pequeñas.

6) Próximamente...

Próximamente...

Proyecto integrador

Próximamente...