Introducción a la programación

Usando Python

G. Sebastián Pedersen

Instituto de Industria
Universidad Nacional de General Sarmiento

Matemática para Economistas III, 1er. cuat. 2019

`https://sebasped.github.io/pythonungs/`

Repaso clase anterior

¿Qué hace el siguiente programa en Python?

```
# m2k_v1.py
# Ingreso las millas (en float para no hacer división entera)
mills = 34.122

# Hago la conversión a kilómetros
kms = 34.122*1.6

# Imprimo por pantalla el resultado
print mills,"millas son",kms,"kilómetros"
```

La salida queda así: 34.122 millas son 54.5952 kilómetros

- ¿Qué problemas tiene?
  - Duplicación de información: 34.122 lo estoy guardando en mills pero después vuelvo a poner 34.112 en kms
- ¿Cómo se podría arreglar?

Repaso clase anterior

Lo mismo sin duplicar información

```
# m2k_v2.py
# Ingreso las millas (en float para no hacer división entera)
mills = 34.122

# Hago la conversión a kilómetros
kms = mills*1.6

# Imprimo por pantalla el resultado
print mills,"millas son",kms,"kilómetros"
```

La salida sigue igual: 34.122 millas son 54.5952 kilómetros

- Duplicación de información corregida: 34.122 lo estoy guardando en mills, y después uso la variable mills y no vuelvo a poner 34.112 en kms

Repaso clase anterior

¿Qué hace el siguiente programa en Python?

```
# m2k_v3.py
# Ingreso las millas
mills = 34.122  # la ingreso float para no hacer división entera

# Hago la conversión a kilómetros
kms = mills*1.6

round(54.592, 2)

# Imprimo por pantalla el resultado
print mills,"millas son",kms,"kilómetros"
```

La salida sigue igual: 34.122 millas son 54.5952 kilómetros

- Lo mismo que la versión 2 (`m2k_v2.py`)
- Bah, casi... ¿Qué pasa con `round(54.592, 2)`?
  - `round(54.592, 2)` redondea 54.592 a dos decimales después de la coma, pero el resultado no se guarda en ningún lado ¿por qué?
- ¿Cómo se podría guardar el resultado del redondeo para utilizarlo después en el programa?
- ¿Qué otro problema hay? Nuevamente duplicación de información: en vez de 54.592 podemos usar `kms` que es la variable que guarda ese valor.

Repaso clase anterior

Guardo el resultado de redondear, y reutilizo el valor de `kms`

```
# m2k_v4.py
# Ingreso las millas
mills = 34.122  # la ingreso float para no hacer división entera

# Hago la conversión a kilómetros
kms = mills*1.6

kmsRedon = round(kms, 2)

# Imprimo por pantalla el resultado
print mills,"millas son",kms,"kilómetros"
```

La salida sigue igual: 34.122 millas son 54.5952 kilómetros

- ¿Qué hace ahora el programa?
  - Esencialmente, todavía lo mismo que la versión 2 (`m2k_v2.py`)...
- ¿Uso en algún lado el valor de `kmsRedon`?
  - No... ¿cómo lo podría usar?
  - En vez de mostrar (imprimir por pantalla) el valor de `kms`, podría mostrar el valor redondeado `kmsRedon`

Repaso clase anterior

Mostrando el valor redondeado de los kilómetros

```
# m2k_v5.py
# Ingreso las millas
mills = 34.122  # la ingreso float para no hacer división entera

# Hago la conversión a kilómetros
kms = mills*1.6

kmsRedon = round(kms, 2)

# Imprimo por pantalla el resultado
print mills,"millas son",kmsRedon,"kilómetros"
```

La salida ahora son kms redondeados: 34.122 millas son 54.6 kilómetros

Utilizando listas

¿Qué hace el siguiente programa?

```
# m2k_v6.py
# Ingreso las millas
mills = [34.122, 17.588, 3.187]  # la ingreso float para no hacer división entera

# Hago la conversión a kilómetros
kms = [mills[0]*1.6, mills[1]*1.6, mills[2]*1.6]

# Imprimo por pantalla el resultado
print mills[0],"millas son",kms[0],"kilómetros"
print mills[1],"millas son",kms[1],"kilómetros"
print mills[2],"millas son",kms[2],"kilómetros"
```

La salida ahora queda:
34.122 millas son 54.5952 kilómetros
17.588 millas son 28.1408 kilómetros
3.187 millas son 5.0992 kilómetros

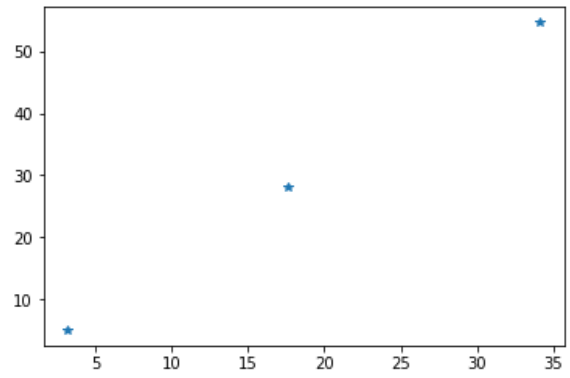
- ¿Cuál es la diferencia fundamental con los anteriores?
  - Esencialmente que ahora son listas de floats y no un único float.
- ¿Tiene información duplicada?
  - El `1.6` aparece 3 veces... ¿podríamos evitar esta duplicación de información?
  - Sí, guardar el `1.6` en una variable, y utilizarlar en vez del `1.6`

Graficando el resultado

```
# m2k_v7.py

# importo la librería para graficar
import matplotlib.pyplot as plt

# Ingreso las millas
mills = [34.122, 17.588, 3.187]  # la ingreso float para no hacer división entera

# Hago la conversión a kilómetros
kms = [mills[0]*1.6, mills[1]*1.6, mills[2]*1.6]

# Imprimo por pantalla el resultado
print mills[0],"millas son",kms[0],"kilómetros"
print mills[1],"millas son",kms[1],"kilómetros"
print mills[2],"millas son",kms[2],"kilómetros"

# Imprimo el resultado en un gráfico
plt.plot(mills,kms)  # plot genera un gráfico X contra Y
plt.show()  # show muestra el gráfico
```

La salida queda así:

34.122 millas son 54.5952 kilómetros
17.588 millas son 28.1408 kilómetros
3.187 millas son 5.0992 kilómetros

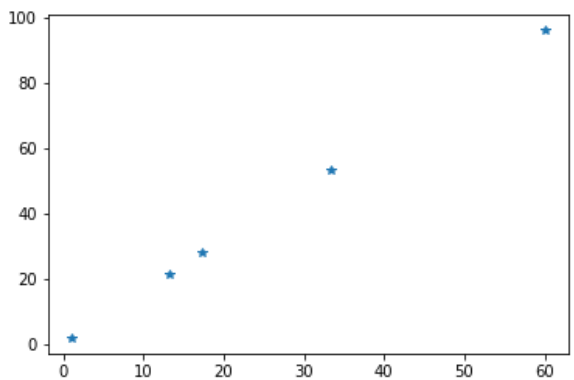


Que funcione para lista de cualquier longitud

¿Qué hace el siguiente programa?

```
# m2k_v8.py
# Ingreso las millas
mills = [1.1, 33.4, 13.2, 60.0, 17.35]  # la ingreso float para no hacer división entera

# guardo la longitud de la lista
long = len(mills)

# Hago la conversión a kilómetros
kms = [0]*long  # lista de ceros de longitud adecuada

i = 0 # variable para posición en la lista
while i < long : # mientras que la posición no se salga de la lista
    kms[i] = mills[i]*1.6  # convertir
    i = i + 1  # avanzar una posición

# Imprimo el resultado en un gráfico
plt.plot(mills,kms,'*')
plt.show()
```

La salida queda así:



- ¿Funciona este programa para lista de cualquier longitud?
- Sí, porque me fijo la longitud de la lista y la recorro toda.

¡Manos a la obra!

Ejercicios:

1. Escribir un script en un archivo d2p.py. El programa tiene que tomar como entrada una lista de dólares, y convertirla a una lista de sus equivalentes en pesos (tomar algún tipo de cambio).
   - El programa tiene que funcionar para lista de cualquier longitud
   - La salida del programa tiene que ser un gráfico Dólares vs. Pesos.
2. Escribir un script en un archivo variacion.py. El programa tiene que tomar como entrada una lista de cotizaciones (en pesos) del dólar, y convertirla a una lista de variaciones porcentuales en la cotización.
   - El programa tiene que funcionar para lista de cualquier longitud.
   - La salida del programa tiene que ser un gráfico de la evolución de las variaciones.
3. Modificar el script del punto 1. y guardarlo en un archivo d2p_v2.py. Ahora la salida del programa además del gráfico, también debe incluir una salida por pantalla indicando para cada monto de dólares su correspondiente monto equivalente en pesos.