

Taller 4

Métodos Computacionales para Políticas Públicas - UROSARIO

Entrega: viernes 11-sep-2020 11:59 PM

[Ivonne Paola Ubaque Galán]

[ivonne.ubaque@urosario.edu.co (<mailto:ivonne.ubaque@urosario.edu.co>)]

Instrucciones:

- Guarde una copia de este *Jupyter Notebook* en su computador, idealmente en una carpeta destinada al material del curso.
- Modifique el nombre del archivo del *notebook*, agregando al final un guión inferior y su nombre y apellido, separados estos últimos por otro guión inferior. Por ejemplo, mi *notebook* se llamaría: mcpp_taller4_santiago_mataallana
- Marque el *notebook* con su nombre y e-mail en el bloque verde arriba. Reemplace el texto "[Su nombre acá]" con su nombre y apellido. Similar para su e-mail.
- Desarrolle la totalidad del taller sobre este *notebook*, insertando las celdas que sea necesario debajo de cada pregunta. Haga buen uso de las celdas para código y de las celdas tipo *markdown* según el caso.
- Recuerde salvar periódicamente sus avances.
- Cuando termine el taller:
 1. Descárguelo en PDF.
 2. Suba los dos archivos (.pdf y .ipynb) a su repositorio en GitHub antes de la fecha y hora límites.

(Todos los ejercicios tienen el mismo valor.)

Zelle, Exercises 6.8 (p. 159):

- True/False: 1-10
- Multiple choice: 2, 3, 6, 7, 10
- Programming Exercises: 1, 3, 4, 11, 12, 13

I. True/False

1. F
2. F

2. .
3. V
4. V
5. V
6. F
7. F
8. V
9. V
10. V

II. Múltiple choice: 2, 3, 6, 7 y 10

2. a
3. a
4. b
5. d
6. c

III. Programing exercises

In [1]: *## 3.1 Primer ejercicio*

```
def cancion():
    print("Old MacDonald had a farm, Ee-igh. Ee-igh, Oh!")

def animal1():
    cancion()
    print("and on that farm he had a cow, Ee-igh. Ee-igh, Oh!")
    cancion()

def animal2():
    cancion()
    cancion()
    print("and on that farm he had a horse, Ee-igh. Ee-igh, Oh!")
    cancion()

def animal3():
    cancion()
    cancion()
    print("and on that farm he had a pig, Ee-igh. Ee-igh, Oh!")
    cancion()

def animal4():
    cancion()
    cancion()
    print("and on that farm he had a chicken, Ee-igh. Ee-igh, Oh!")
    cancion()

def animal5():
    cancion()
    cancion()
    print("and on that farm he had a cat, Ee-igh. Ee-igh, Oh!")
    cancion()

def cancionfinal():
    animal1()
    print()
    animal2()
    print()
    animal3()
    print()
    animal4()
    print()
    animal5()
    print()
```

In [2]: `cancionfinal()`

```
Old MacDonald had a farm, Ee-igh. Ee-igh, Oh!
Old MacDonald had a farm, Ee-igh. Ee-igh, Oh!
and on that farm he had a horse, Ee-igh. Ee-igh, Oh!
Old MacDonald had a farm, Ee-igh. Ee-igh, Oh!

Old MacDonald had a farm, Ee-igh. Ee-igh, Oh!
Old MacDonald had a farm, Ee-igh. Ee-igh, Oh!
and on that farm he had a pig, Ee-igh. Ee-igh, Oh!
Old MacDonald had a farm, Ee-igh. Ee-igh, Oh!

Old MacDonald had a farm, Ee-igh. Ee-igh, Oh!
Old MacDonald had a farm, Ee-igh. Ee-igh, Oh!
and on that farm he had a chicken, Ee-igh. Ee-igh, Oh!
Old MacDonald had a farm, Ee-igh. Ee-igh, Oh!

Old MacDonald had a farm, Ee-igh. Ee-igh, Oh!
Old MacDonald had a farm, Ee-igh. Ee-igh, Oh!
and on that farm he had a cat, Ee-igh. Ee-igh, Oh!
Old MacDonald had a farm, Ee-igh. Ee-igh, Oh!
```

3.2 Segundo ejercicio. N° 3

Ejercicios de programación capítulo 3. Ejercicio 1 Escriba un programa que calcule el volumen y área de superficie de una esfera, a partir de su radio, dado como entrada. Tenga en cuenta las formulas dadas en el ejercicio.

In [3]: `from math import pi`

```
r = float(input('Por favor, escriba el radio de la esfera: '))

def spherearea(r):
    area = 4*pi*r**2
    return area

def volume(r):
    volumen = 4/3*pi*r**3
    return volumen

radio = r
print("El área de la esfera es", spherearea(radio))
print("el volumen de la esfera es", volume(r))
```

```
Por favor, escriba el radio de la esfera: 10
El área de la esfera es 1256.6370614359173
el volumen de la esfera es 4188.790204786391
```

3.3. Tercer Ejercicio. N° 4.

Escriba definiciones para las siguientes dos funciones:

`sumN(n)` devuelve la suma de los primeros `n` números naturales.

sumNCubes(n): devuelve la suma de los cubos de los primeros n números naturales.

Luego use estas oraciones en un programa que solicite el uso de n e imprima la suma de los primeros n números naturales y la suma de los cubos de los primeros n números naturales.

```
In [7]: ### Devolución de la suma de los primeros n números naturales.

def sumar(n):
    return(n*(n+1))/2

numeroentero = float(input('Por favor, escriba un número entero: '))
suma = sumar(numeroentero)

print("Su resultado es: ", suma)
```

Por favor, escriba un número entero: 99
Su resultado es: 4950.0

```
In [8]: ### Devolución de la sumatoria de cubos del número solicitado

def suma_cubos(n):
    suma = 0
    n -= 1

    while n > 0:
        suma += n**3
        n -= 1

    return suma

numeroentero = float(input('Por favor, escriba un número entero para calcularle la sumatoria de cubos: '))
sumacubos = suma_cubos(numeroentero)

print("Su resultado es: ", sumacubos)
```

Por favor, escriba un número entero para calcularle la sumatoria de cubos: 99
Su resultado es: 23532201.0

3.4. Ejercicio. N° 11. Escribe una prueba de una función para cumplir con esta especificación:

squareEach (nums) nums es una lista de números. Modifica la lista, elevando al cuadrado cada entrada

```
In [10]: import numpy as np

def potencia_cuadrado(arreglo):
    iterador = np.nditer([arreglo, None])

    for a, b in iterador:
        b[...] = a**2

    return iterador.operands[1]

potencia_cuadrado([5.4, 1, 45, 89.5, 100, 2])
```

```
Out[10]: array([2.91600e+01, 1.00000e+00, 2.02500e+03, 8.01025e+03, 1.00000e+04,
                4.00000e+00])
```

3.5 Ejercicio. N° 12. Escribe una prueba de una función para cumplir con esta especificación:

Devuelva la suma de los números de una lista

```
In [9]: def suma_lista(lista):
        suma = 0

        for numero in lista:
            suma += numero

        return suma

lista_ensayo = [47.8, 1, 108, 99, 2.24]

print(suma_lista (lista_ensayo))

258.04
```