

## Taller 4

Métodos Computacionales para Políticas Públicas - UROSario

Entrega: viernes 1-mar-2019 11:59 PM

**\*\*Juan Sebastián Valbuena\*\***  
  
juans.valbuena@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

1. True/False: 1-10

- 1- Programmers rarely their own functions. False
- 2- A function may only be called at one place in a program. False
3. Information can be passed into a function through parameters. True
4. Every Python returns some calue. True.
- 5- In Python, some parameters are passed by reference. False
- 6- In Python, a function can return only one value. False.
7. Python functions can never modify a parameter. False
- 8- One reason to use functions is to reduce code duplication. True
- 9- Variables defined in a function are local to that function. True.
- 10- It's a bad idea to define new functions if it makes a program longer. True.

- Multiple choice: 2, 3, 6, 7, 10
  1. b.
  2. a.
  3. a.
  4. d.

• Programming Exercises: 1, 3, 4, 11, 12, 13

## 1

```
In [ ]: def animal(nombre, sonido):
        print('Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!')
        print('And on that farm he had a ' + nombre + ', Ee-igh, Eei-gh, Oh!')
        print('With a ' + sonido + ', ' + sonido + ' here and a ' + sonido + ', ' + sonido + ' there.')
        print('Here a ' + sonido + ', there a ' + sonido + ', everywhere a ' + sonido + ', ' + sonido +
        ',.')
        print('Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!\n')

        def main():
            animal('cow', 'moo')
            animal('sheep', 'baah')
            animal( "duck", "quack")
            animal('pig', 'oink')
            animal("dog", "woof")

main()
```

## 3

```
In [ ]: from math import*
def sphereArea(radius):
    A = 4*pi*radius**2
    return A
def sphereVolume(radius):
    V = (4/3)*pi*radius**3
    return V
def main():
    print("El programa encuentra el volumen y el área de una esfera.\n")
    r = eval(input("¿Cuál es el radio de la esfera?(in.): "))
    V = sphereVolume(r)
    A = sphereArea(r)

print("\nEl area y volumen de radio, {2} , ".format(A,V,r))
print("is {0:0.3f}in^2 y {1:0.3f}in^3.".format(A,V,r))

main()
```

```
In [ ]: from math import*
def sphereArea(radius):
    area = 4 * pi * radius**2
    return area
def sphereVolume(radius):
    volume = 4/3 * pi * radius**3
    return volume

def main():
    print("El programa encuentra el volumen y el área de una esfera")
    r = eval(input("¿Cuál es el radio de la esfera?"))
    print()
    print("El volumen es {0:0.2f}".format(sphereVolume(r)))
    print("El area es {0:0.2f}".format(sphereArea(r)))
main()
```

El programa encuentra el volumen y el área de una esfera

## 4

```
In [ ]: def SumN(n):
        sum = 0
        for num in range(1, n + 1):
            sum = sum + num
        return sum

def sumNCubes(n):
    sum = 0
    for num in range(1, n + 1):
        sum = sum + num**3
    return sum

def main():
    print("Este programa encuentra la suma de los primeros n números naturales.")
    print()
    print("Este programa también encuentra la suma de cubos de los primeros n números naturales")
    n = eval(input("¿Cuál es el número?"))
    print("suma de los primeros n números naturales",sumNCubes(n))
    print("Suma de cubos de los primeros n números naturales.", SumN(n))

main()
```

Este programa encuentra la suma de los primeros n números naturales.

Este programa también encuentra la suma de cubos de los primeros n números naturales

## 11

```
In [13]: def squareEach(numeros):
        for i in range(len(numeros)):
            numeros[i] = numeros[i]**4

def main():
    print("El programa enumera una lista de números:")
    numeros = [2,4,6]
    squareEach(numeros)
    listNum = [2, 4, 6]

    print(listNum)
    print(numeros)

main()
```

El programa enumera una lista de números:  
[2, 4, 6]  
[16, 256, 1296]

## 12

```
In [2]: def sumList(lista):
        total = 0
        for i in range(len(lista)):
            total = int(lista[i]) + total
        return total
def main():
    print("El Programa encuentra suma de números en una lista")
    test = [5, 5, 20, 13, 14, 16, 18, 20]
    print("\nInput:", test)
    sumList(test)
    print("\nOutput:", sumList(test))

main()
```

El Programa encuentra suma de números en una lista

Input: [5, 5, 20, 13, 14, 16, 18, 20]

Output: 111

## 13

```
In [7]: def toNumber(strlist):
        for i in range(len(strlist)):
            strlist[i] = eval(strlist[i])
def main():
    print("Este programa convierte una lista de strings a números")
    numbers = ["3","2","1"]
    print(numbers)
    toNumber(numbers)
    print(numbers)

main()
```

Este programa convierte una lista de strings a números  
['3', '2', '1']