Taller 4

Métodos Computacionales para Políticas Públicas - URosario

Entrega: viernes 28-feb-2020 11:59 PM

[Monica Alejandra Robayo Gonzalez]

[monica.robayo@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_matallana
- 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 = False

2 = False

3 = True

4 = True

5 = False

6 = False

7 = False

8 = True

9 = True

10 = False

2. Multiple choice: 2, 3, 6, 7, 10

2 = (a) def

3 = (a) return

6 = (a) by value

7 = (d) to demonstrate intellectual superiority

10 = (a) mutable

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

1. write a program to print the lyrics of the song "Old MacDonald". Your program should print the lyrics for five different animals, similar to the example verse below.

Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!

And on that farm he had a cow, Ee-igh, Ee-igh, Oh!

With a moo, moo here and moo, moo there.

Here a moo, there a moo, everywhere a moo, moo.

Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!

In [156]:

```
def animal (tipo_animal,sonido):
    print ("Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!")
    print('And on that farm he had a ' + tipo_animal + ', 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 + '.')

def main():
    animal("cow", "moo")
    print ()
    animal("cat", "miau")
    print ()
    animal("dog", "woof")
    print ()
    animal("duck", "quack")
    print ()
    animal ("bird", "cucurr")
    print ()

main ()
```

```
Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!
And on that farm he had a cow, Ee-igh, Eei-gh, Oh!
With a moo, moo here and a moo, moo there.
Here a moo, there a moo, everywhere a moo, moo.

Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!
And on that farm he had a cat, Ee-igh, Eei-gh, Oh!
With a miau, miau here and a miau, miau there.
Here a miau, there a miau, everywhere a miau, miau.

Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!
And on that farm he had a dog, Ee-igh, Eei-gh, Oh!
With a woof, woof here and a woof, woof there.
Here a woof, there a woof, everywhere a woof, woof.

Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!
And on that farm he had a duck, Ee-igh, Eei-gh, Oh!
With a quack, quack here and a quack, quack there.
Here a quack, there a quack, everywhere a quack, quack.

Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!
And on that farm he had a bird, Ee-igh, Eei-gh, Oh!
With a cucurr, cucurr here and a cucurr, cucurr there.
Here a cucurr, there a cucurr, everywhere a cucurr, cucurr.
```

3. Write definitions for these functions:

sphereArea(radius) Returns the surface area of a sphere having the radius.

sphereVolume(radius) Returns the volume of a sphere having the given radius.

Use your functions to solve Programming Exercise 1 from Chapter 3.

Write a program to calculaate the volume and surface area of a sphere from its radius, given as input. Here are some formulas that might be useful:

V= 4/(3*(pi)*(r^3))

A= 4*(pi)*(r^2)

In [1]:

```
#sphereArea(radius) : dado un nivel de radio da el valor del area de la esfera

#sphereVolume(radius) : dado un nivel de radio retorna el valor del volumen de la esfera
```

Volume and surface area of a sphere from its radius

In [63]:

```
# Importamos la libreria de matemáticas para el valor pi.
import math
pi = math.pi

# Funcion para calcular el volumen de la esfera:
def sphereVolume(radius):
    V = (4 / 3) * pi *radius**3
    return V

# Funcion para calcular el area de la superficie de la esfera:
def sphereArea(radius):
    A = 4 * pi *  radius**2
    return A
```

In [67]:

```
print ("el volumen de una esfera dado un radio de 2, es:")
sphereVolume(2)
```

```
el volumen de una esfera dado un radio de 2, es:
```

Out[67]:

```
33.510321638291124
```

In [68]:

```
print ("el area de una esfera dado un radio de 2, es:")
sphereArea(2)
```

```
el area de una esfera dado un radio de 2, es:
```

Out[68]:

```
50.26548245743669
```

4. Write definitions for the following two functions:

sumN(n) returns the sum of the first n natural numbers.

sumNCubes(n) returns the sum of the cubes of the first n natural numbers.

Then use these functions in a program that prompts a user for n and prints out the sum of the first n natural numbers and the sum of the cubes of the first n natural numbers.

In [74]:

```
# sumN(n): suma los primeros n Numeros naturales de una lista.
# sumNCubes(n): suma de los cubos de los primeros N numeros naturales de una lista.
```

In [69]:

```
def sumN(n):
    numero=0
    for i in range(1,n+1):
        numero = numero + i
    return numero
```

In [72]:

```
sumN(10)
```

Out[72]:

```
55
```

In [70]:

```
def sumNCubes(n):
    numero=0
    for i in range(1,n+1):
        numero = numero + i**3
    return numero
```

In [73]:

```
sumNCubes(10)
```

Out[73]:

```
3025
```

11. Write and test a function to meet this specification.

sumList(nums) nums is a list of numbers. Modifies the list by squaring each entry.

In [100]:

```
nums = [2,4,6,8,10]

def squaring (nums):
    print("retorna cada numero de la lista al cuadrado")
    for i in list(range(len(nums))):
        nums [i] = nums[i]**2

    return nums
```

In [103]:

```
squaring (nums)
```

```
retorna cada numero de la lista al cuadrado
```

Out[103]:

```
[4, 16, 36, 64, 100]
```

12. Write and test a function to meet this specification.

sumList(nums) nums is a list of numbers. Returns the sum of the numbers in the list.

In [119]:

```
nums = [1,2,4,6,8,10]

def sumList (nums):
    print("Returns the sum of the numbers in the list.")
    for i in list(range(1,len(nums))):
        nums [i] = nums[i] + nums[i-1]

    return nums
```

In [120]:

```
sumList (nums)
```

```
Returns the sum of the numbers in the list.
```

Out[120]:

```
[1, 3, 7, 13, 21, 31]
```

13. Write and test a function to meet this specification.

toNumbers(strList) strList is a list of strings, each of which represents a number. Modifies each entry in the list by converting it to a number.

In [126]:

```
strList = [ "1", "2", "3"]

def toNumbers(strList):
    print ("Retornar cada string en un numero")
    for i in list(range(len(strList))):
        strList [i] = int(strList[i])

    return strList
```

In [127]:

```
toNumbers(strList)
```

```
Retornar cada string en un numero
```

Out[127]:

```
[1, 2, 3]
```