mcpp_taller4_juan_salgado

August 30, 2019

1 Taller 4

Métodos Computacionales para Políticas Públicas - URosario
Entrega: viernes 30-ago-2019 11:59 PM
Juan Camilo Salgado Ramírez juanca.salgado@urosario.edu.co

1.1 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.)

1.2 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

1.2.1 1) True/False: 1-10

1) False
2) False
3) True
4) True: si no retorna explícitamente algo, el return sera none (por lo que si retorna algo)
5) False
6) False
7) False
8) True
9) True
10) False

1.2.2 2) Multiple choice

Ex. #2 - A Python function definition begins with:

a) def

Ex. #3 - A function can send output back to the program with a(n):

a) return

Ex. #6 - In Python, actual parameters are passed to function

a) by value

Ex. #7 - Which of the following is not a reason to use functions?

d) to demonstrate intellectual superiority

Ex. 10 - A function can modify the value of an actual parameter only if it's

a) mutable

1.2.3 3) Programming exercises

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 a moo, moo there. Here a moo, there a moo, everywhere a moo, moo. Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!

```
[1]: animals = ["cat", "dog", 'cow', 'snake', 'lion']
     sound = ['miau', 'guau', 'moo', 'sss', 'waurr']
     for i in range(len(animals)):
         print("Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!")
         print('And on that farm he had a ' + animals[i] +', Ee-igh, Ee-igh, Oh!')
         print('With a ' + sound[i] + ', ' + sound[i] + ' here and a ' + sound[i] +
         ↪', ' + sound[i] + ' there.')
```

```python
    print('Here a ' + sound[i] + ', there a ' + sound[i] + ', everywhere a ' +␣
 ↪sound[i] + ', ' + sound[i])
    print('Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!')
    print("")
```

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

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

Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!
And on that farm he had a snake, Ee-igh, Ee-igh, Oh!
With a sss, sss here and a sss, sss there.
Here a sss, there a sss, everywhere a sss, sss
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 lion, Ee-igh, Ee-igh, Oh!
With a waurr, waurr here and a waurr, waurr there.
Here a waurr, there a waurr, everywhere a waurr, waurr
Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!
```

3) Write definitions for these functions: - sphereArea(radius): Returns the surface area of a sphere having the given radius. - sphereVolume(radius): Returns the volume of a sphere having the given radius.

```python
pi = 3.14159265359

def sphereArea(radius, pi = pi):
    a = 4*pi*radius**2
    return a

def sphereVolume(radius, pi = pi):
    v = 4/(3*pi*radius**3)
```

```
    return v

radio = int(input('Digite radio: '))

print(radio)
print(sphereArea(radio))
print(sphereVolume(radio))
```

```
Digite radio: 3
3
113.09733552924
0.01571900672512443
```

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.

[3]:
```python
def sumN(n):
    suma = 0
    for i in range(1,n+1):
        suma = suma + i
    return suma

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

num = int(input('Digite número: '))
print('suma ' + str(sumN(num)))
print('suma al cubo ' + str(sumNCubes(num)))
```

```
Digite número: 7
suma 28
suma al cubo 784
```

11) Write and test a function to meet this specification: - squareEach(nums): *nums* is a list of numbers. Modifies the list by squaring each entry.

[4]:
```python
def squareEach(nums):
    for i in range(0, len(nums)):
        nums[i] = nums[i]**2

lista = [4,2,3,4,9]
print('Inicial: ', lista)
squareEach(lista)
print('Final: ', lista)
```

```
Inicial:  [4, 2, 3, 4, 9]
Final:  [16, 4, 9, 16, 81]
```

4

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.

```python
[5]: def sumList(nums):
         suma = 0
         for i in range(0, len(nums)):
             suma = suma + nums[i]
         return suma

     lista = [4,2,3,4,9]
     print('Lista: ', lista)
     sumList(lista)
     print('Suma: ', sumList(lista))
```

```
Lista:  [4, 2, 3, 4, 9]
Suma:   22
```

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.

```python
[6]: def toNumbers(strList):
         for i in range(0, len(strList)):
             strList[i] = int(strList[i])

     lista = ['6','2','3','4','9']
     print('Lista: ', lista)
     toNumbers(lista)
     print('Lista convertida: ', lista)
```

```
Lista:  ['6', '2', '3', '4', '9']
Lista convertida:  [6, 2, 3, 4, 9]
```