mcpp_taller_4_juan_munoz

August 28, 2019

1 Taller 4

Métodos Computacionales para Políticas Públicas - URosario

Entrega: viernes 30-ago-2019 11:59 PM

jJuan Sebastián Muñoz Vargas jsebastianmvargas@gmail.com

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

1.3 Solución del taller

1.3.1 True/false

1. Programmers rarely define 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 function returns some value. 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 FALSE

1.3.2 Multiple choice 2, 3, 6, 7, 10

2. A Python function definition begins with

a) def

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

a) return

6. In Python, actual parameters are passed to functions

a) by value

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

d) to demonstrate intellectual superiority

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

a) mutable

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

[13]:
```
#Ejercicio 1
for i in range (0, 5):
    if i == 0:
        animal = "cow"
        sound = "moo"
```

```
    elif i == 1:
        animal = "cat"
        sound = "meow"
    elif i == 2:
        animal = "lion"
        sound = "grr"
    elif i == 3:
        animal = "dog"
        sound = "woof"
    elif i == 4:
        animal = "snake"
        sound = "ZZZ"
    print ("Old MacDonald had a farm, Ee-igh, Ee-igh, Oh! \n And on that farm␣
↪he had a " + animal + ", Ee-igh, Ee-igh, Oh! \n With a " + sound + ", " +␣
↪sound + " here and a " + sound + ", " + sound + " there.\n Here a " + sound␣
↪+  ", there a " + sound + ", everywhere a " + sound + ", " + sound + ". \n␣
↪Old MacDonald had a farm, Ee-igh, Ee-igh, Oh! \n \n" )
```

```
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 cat, Ee-igh, Ee-igh, Oh!
 With a meow, meow here and a meow, meow there.
 Here a meow, there a meow, everywhere a meow, meow.
 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 grr, grr here and a grr, grr there.
 Here a grr, there a grr, everywhere a grr, grr.
 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 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!


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 ZZZ, ZZZ here and a ZZZ, ZZZ there.
Here a ZZZ, there a ZZZ, everywhere a ZZZ, ZZZ.
Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!
```

[28]:
```python
#Ejercicio 3
import math
def sphereArea (radius):
    '''
    #Returns the surface area of a sphere having the given radius
    '''
    V = (((4/3)*math.pi) * radius ** 3)
    return V
def sphereVolume (radius):
    '''
    #Returns the volume of a sphere having the given radius
    '''
    A = ((4 * math.pi) * (radius ** 2))
    return A
print(sphereArea (2))
print (sphereVolume (2))
```

```
33.510321638291124
50.26548245743669
```

[55]:
```python
#Ejercicio 4
n = input ("Seleccione un número entero ")
n = int (n)

def sumN (n):
    '''
    returns the sum of the first n natural numbers
    '''
    r1 = 0
    for i in range (n):
        r1 = r1 + (i + 1)
    return r1
def sumNCubes (n):
    '''
    returns the sum of the cubes of the first n natural numbers
    '''
    r2 = 0
    for i in range (n):
        r2 = (r2 + (i + 1) ** 3)
    return r2
```

```
z = print ("La suma de todos los números de 1 hasta", n , "es" , sumN (n))
C = print ("La suma de todos los cubos de 1 hasta", n , "es" , sumNCubes (n))
```

```
Seleccione un número entero 2
La suma de todos los números de 1 hasta 2 es 3
La suma de todos los cubos de 1 hasta 2 es 9
```

[79]:
```
#Ejercicio 11
def squareEach (nums):
    '''
    Inputs:
        nums = a list of numbers.
    Outputs
        A list where each entry is squared.
    '''
    list2 = []
    for i in nums:
        list2.append (i ** 2)
    return list2
Lista = [1, 2, 3, 5]
squareEach (Lista)
```

[79]:
```
[1, 4, 9, 25]
```

[80]:
```
#Ejercicio 12
def sumList (nums):
    '''
    Inputs:
        nums = a list of numbers.
    Outputs
         The sum of the numbers in the list.
    '''
    suma = 0
    for i in nums:
        suma = suma + i
    return suma
Lista = [1, 2, 3, 5]
sumList (Lista)
```

[80]:
```
11
```

[131]:
```
#Ejercicio 13
def toNumbers (strList):
    '''
    Inputs:
        strList = a list of strings, each of which represents a number.
    Outputs
         Modifies each entry in the list by converting it to a number.
    '''
```

```
    finalList = []
    for i in strList:
        for s in i.split(","):
            s = eval (s)
            finalList.append (s)
    return finalList
Lista = "164512"
Z = toNumbers (Lista)
print (Z)
```

```
[1, 6, 4, 5, 1, 2]
```