

mcpp_taller4_Laura_Becerra

August 28, 2016

1 Taller 4

Métodos Computacionales para Políticas Públicas - URosario

Entrega: viernes 2-sep-2016 11:59 PM

[Laura Becerra] [l.becerra52@uniandes.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

True/False:

1. Programmers rarely define their own functions: Falso, los programadores usan funciones para reducir la duplicación de código y hacer un programa más ordenado.

2. A function may only be called at one place in a program: Falso, una función puede llamarse varias veces en un programa.
3. Information can be passed into a function through parameters: Verdadero
4. Every Python function returns some value: Verdadero, siempre y cuando se ponga return porque si no aparece NONE.
5. In Python, some parameters are passed by reference: Falso, Python pasa parametros "by value."
6. In Python, a function can return only one value: Falso, las funciones pueden dar más de un valor.
7. Python functions can never modify a parameter: Falso, las funciones pueden modificar objetos mutables como listas.
8. One reason to use functions is to reduce code duplication: Verdadero.
9. Variables defined in a function are local to that function: Verdadero como lo vimos en clase.
10. It's a bad idea to define new functions if it makes a program longer: Falso, las funciones pueden ser de ayuda para organizar el código así el programa resulte más largo.

Selección multiple:

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

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!

```
In [5]: def animals(animal, sound):
        print ("Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!")
        print ("And on the farm he had a ", animal + ", Ee-igh, Ee-igh, Oh!")
        print ("With a", sound + ", ", sound, "here and a" , sound + ", ", sound +
        print ("Here a", sound + ", there a", sound + ", everywhere a", sound +
```

```

    print ("Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!")
    print
    return animals

def oldMac():
    animals("cow", "moo")
    animals("pig", "oink")
    animals("horse", "neigh")
    animals("dog", "woof")
    animals("cat", "meow")

oldMac()

```

Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!
 And on the 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 the farm he had a pig, Ee-igh, Ee-igh, Oh!
 With a oink, oink here and a oink, oink there.
 Here a oink, there a oink, everywhere a oink, oink.
 Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!
 Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!
 And on the farm he had a horse, Ee-igh, Ee-igh, Oh!
 With a neigh, neigh here and a neigh, neigh there.
 Here a neigh, there a neigh, everywhere a neigh, neigh.
 Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!
 Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!
 And on the 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 the 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!

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. Use your functions to solve Programming Exercise 1 from Chapter 3. $V = 4/3\pi(r^3)$ $A = 4\pi(r^2)$

```

In [7]: import math

def sphereArea(radius):

```

```

        area = math.pi*4*(radius**2)
        return area

def sphereVolume(radius):
    volume = (4/3)*math.pi*(radius**3)
    return volume

def main():
    print('Programa para calcular el area y volumen de una esfera.')
    ingresoradio = eval(input('Por favor entre el radio de su esfera: '))
    area = sphereArea(ingresoradio)
    vol = sphereVolume(ingresoradio)
    print('\ncon radio de {}, el área de la esfera es {:.2f} y el volumen es {:.2f}'.format(ingresoradio, area, vol))

main()

```

Programa para calcular el area y volumen de una esfera.
 Por favor entre el radio de su esfera: 2

con radio de 2, el área de la esfera es 50.27 y el volumen es 33.51.

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 [1]: def sumN(n):
        result = 0
        for i in range(n+1):
            result += i
        return result

def sumNCubes(n):
    result = 0
    for i in range(n+1):
        result += i**2
    return result

def prompt_user():
    n = int(input("por favor ingrese n:"))
    print("la suma de los primeros n numeros es:", sumN(n))
    print("la suma de los primeros n cubos es:", sumNCubes(n))

prompt_user()

```

por favor ingrese n:4
 la suma de los primeros n numeros es: 10

la suma de los primeros n cubos es: 30

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.

```
In [2]: def squareEach(nums):
        entry = 0
        for i in nums:
            nums[entry] = i**2
            entry = entry+1

        def main():
            print('un programa para obtener numeros al cuadrado.')
            nums = input('por favor ingrese varios numeros separados por una coma: ')

            nums = nums.split(',')

            entry = 0
            for i in nums:
                nums[entry] = int(i)
                entry = entry+1

            squareEach(nums)

            print('\neste es el resultado: ',nums)
        main()
```

un programa para obtener numeros al cuadrado.
por favor ingrese varios numeros separados por una coma: 5,6,7

este es el resultado: [25, 36, 49]

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 [3]: def sumList(nums):
        total = 0
        for i in nums:
            total = total+i
        return total

        def main():
            print('un programa para sumar los números dados.')
            nums = input('por favor ingrese varios números separados por coma: ')

            nums = nums.split(',')

```

```

entry = 0
for i in nums:
    nums[entry] = int(i)
    entry = entry+1

sumTotal = sumList(nums)

print('\nla suma de los números es:{}.'.format(sumTotal))
main()

```

un programa para sumar los números dados.
por favor ingrese varios números separados por coma: 4,5,6

la suma de los números es:15.

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 [2]: def toNumbers(strList):
        return [int(num) for num in strList]

```

```

a=['1','2','3']

```

```

b=toNumbers(a)
b

```

```

Out[2]: [1, 2, 3]

```