

Taller 4

Métodos Computacionales para Políticas Públicas - URosario

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

****Francisco Monsalve****

francisco.monsalve@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

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

Multiple choice

- 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 its
 - a) mutable

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

1. Print "Old MacDonald" for 5 different animals

```
In [1]: sound = ["moo", "oink", "quak", "neigh", "baa"]
```

```
animals = ["cow", "pig", "duck", "horse", "lamb"]
```

Old MacDonald had a farm, Ee-igh, Ee-igh, Oh! And on his farm he had a X, Ee-igh, Ee-igh, Oh!
With a Y Y here and a Y Y there. Here a Y, there a Y, everywhere a Y, Y. Old MacDonald had a
farm, Ee-igh, Ee-igh, Oh!

```
In [4]: for i in range(5):  
        print("Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!", "\n", "And o  
n his 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.", "\n", "Here a", sound[i], ",", "there a ", sound[i],  
",", "everywhere a ", sound[i], ",", sound[i], ".", "\n", "Old MacD  
onald had a farm, Ee-igh, Ee-igh, Oh!", "\n")
```

Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!
And on his 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 his 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 his farm he had a duck Ee-igh, Ee-igh, Oh!
with a quak , quak here and a quak , quak there.
Here a quak , there a quak , everywhere a quak , quak .
Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!

Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!
And on his 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 his farm he had a lamb Ee-igh, Ee-igh, Oh!
with a baa , baa here and a baa , baa there.
Here a baa , there a baa , everywhere a baa , baa .
Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!

3. Write definitions for 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

```
In [5]: pi = 3.141592
```

```
In [6]: def sphereArea(r):  
        a = (4)*pi*(r**2)  
        return a
```

```
In [7]: sphereArea(5)
```

```
Out[7]: 314.1592
```

```
In [8]: def sphereVolume(r):  
        a = (4/3)*pi*(r**3)  
        return a
```

```
In [9]: sphereVolume(8)
```

```
Out[9]: 2144.6601386666666
```

4. Write definitions

- `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 then the sum of the cubes of the first n natural numbers. (Natural numbers= non negative integers)

```
In [10]: def sumN(n):  
        s=0  
        for i in range(1,n+1):  
            s=s+i  
        return s
```

```
In [11]: sumN(9)
```

```
Out[11]: 45
```

```
In [12]: def sumNCubes(n):  
        sc = 0  
        for i in range(1,n+1):  
            sc= sc + (i**3)  
        return sc
```

```
In [13]: sumNCubes(3)
```

```
Out[13]: 36
```

11. Write and test a function to meet this specification

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



```
In [14]: nums =[1,3,2,4,6,7,8,3,5,2,1,1]  
        nums2 = [1,2,35,7,5,3,7,8,9]
```

```
In [15]: numsq = []
```

```
def squareEach(nums):  
    for i in nums:  
        sq = i**2  
        numsq.append(sq)  
    return numsq
```

In [16]: squareEach(nums)

Out[16]: [1, 9, 4, 16, 36, 49, 64, 9, 25, 4, 1, 1]

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 [17]: def sumList(nums):  
        suml = sum(nums)  
        return suml
```

In [18]: sumList(nums2)

Out[18]: 77

```
In [19]: # Definir una función, sin usar "sum"  
def sumList1(nums):  
    suml = 0  
    for i in nums:  
        suml = suml + i  
    return suml
```

In [20]: sumList1(nums2)

Out[20]: 77

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 [21]: strList=["1","2","3","4","5","10","20"]
```

```
In [22]: type(strList)
```

```
Out[22]: list
```

```
In [23]: type(strList[1])
```

```
Out[23]: str
```

```
In [24]: def toNumbers(strList):  
         for i in range(len(strList)):  
             strList[i]=int(strList[i])  
         return strList
```

```
In [25]: intlist=["1","2","3","4","5","10","20"]  
         toNumbers(intlist)
```

```
Out[25]: [1, 2, 3, 4, 5, 10, 20]
```

```
In [27]: type(intlist[1])
```

```
Out[27]: int
```

```
In [ ]:
```