mcpp_taller4_monica_gasca

September 2, 2016

1 Taller 4

Métodos Computacionales para Políticas Públicas - URosario

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

[Mónica Gasca] [monicagascarojas@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 Ejercicio verdadero/falso

1.False
2.False, it may be called multiple times.
3.True, parameters allow functions to have changeable parts.
4.False
5.False, parameters are always passed by values in python.
6.False, may return multiple values.
7.False, functions can communicate back to the calling program by making changes to the parameters.
8.True
9.True
10.False

1.4 Ejercicio opción múltiple

2.a
3.a
6.a
7.d
10.a

1.5 Ejercicios de programación

1.5.1 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.

```
In [37]: def song ():
             print ("Old MacDonald had a farm, Ee-igh, Ee-igh, Oh! ")

         def songcow():
             song()
             print ("And on the farm he had a cow, Ee-igh, Ee-igh, Oh!")
             print ("With a moo, here and a moo, moo there.")
             print ("Here a moo, there a moo, everywhere a moo, moo.")
             song()

         def songpig():
             song()
             print ("And on the farm he had a pig, Ee-igh, Ee-igh, Oh!")
             print ("With a oink, here and a oink, oink there.")
             print ("Here a oink, there a oink, everywhere a oink, oink.")
             song()

         def songdog():
             song()
             print ("And on the farm he had a dog, Ee-igh, Ee-igh, Oh!")
             print ("With a guau, here and a guau, guau there.")
             print ("Here a guau, there a guau, everywhere a guau, guau.")
```

```python
    song()

def songcat():
    song()
    print ("And on the farm he had a cat, Ee-igh, Ee-igh, Oh!")
    print ("With a miau, here and a miau, miau there.")
    print ("Here a miau, there a miau, everywhere a miau, miau.")
    song()

def songsheep():
    song()
    print ("And on the farm he had a sheep, Ee-igh, Ee-igh, Oh!")
    print ("With a beeeh, here and a beeeh, beeeh there.")
    print ("Here a beeeh, there a beeeh, everywhere a beeeh, beeeh.")
    song()

def main():
    songcow()
    songpig()
    songdog()
    songcat()
    songsheep()

main()
```

```
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, 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, 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 dog, Ee-igh, Ee-igh, Oh!
With a 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 the farm he had a cat, Ee-igh, Ee-igh, Oh!
With a 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 the farm he had a sheep, Ee-igh, Ee-igh, Oh!
With a beeeh, here and a beeeh, beeeh there.
```

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

```
In [45]: 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 + ", here and a ", sound + ", ", sound + " the
             print ("Here a ", sound + ", there a ", sound + ", everywhere a ", sou
             print ("Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!" )

         def song():
             animals ("cow","moo")
             animals ("pig", "oink")
             animals ("cat", "miau")
             animals ("dog", "guau")
             animals ("sheep", "beeeh")

         song()
```

```
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, 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, 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  cat, Ee-igh, Ee-igh, Oh!
With a  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 the farm he had a  dog, Ee-igh, Ee-igh, Oh!
With a  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 the farm he had a  sheep, Ee-igh, Ee-igh, Oh!
With a  beeeh, here and a  beeeh,  beeeh there.
Here a  beeeh, there a  beeeh, everywhere a  beeeh,  beeeh.
Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!
```

1.5.2 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.

```
In [1]: pi=3.14159265358979323846

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

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

        def main():
            print ("Calculadora de area y volumen")
            radio= eval (input("Ingrese el radio"))
            area = sphereArea(radio)
            volumen= sphereVolume(radio)
            print ("Con un radio de", radio)
            print ("El área de la esfera es", area)
            print ("El volumen de la esfera es", volumen)

        main ()
```

```
Calculadora de area y volumen
Ingrese el radio5
Con un radio de 5
El área de la esfera es 314.1592653589793
El volumen de la esfera es 523.5987755982989
```

1.5.3 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.

```
In [4]: def sumN(n):
            resultado = 0
            for i in range(n + 1):
                resultado += i
            return resultado

        def sumNcubes(n):
            resultado = 0
            for i in range (n + 1):
                resultado += i**3
```

```
    return resultado

def resultados():
    n = int (input("Ingrese un n: "))
    print ("La suma de los primeros n números es ",sumN(n))
    print ("la suma de los cubos de los primeros n números es",sumNcubes(n)

resultados()
```

```
Ingrese un n: 8
La suma de los primeros n números es  36
la suma de los cubos de los primeros n números es 1296
```

1.5.4 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 [27]: def squareEach(nums):
             resultado = 0
             for i in nums:
                 nums [resultado] = i**2
                 resultado = resultado+1

         def main():
             print ("Números al cuadrado")
             nums = input("ingrese una lista de números separados por comas: ")
             nums= nums.split(",")

             resultado= 0
             for i in nums:
                 nums [resultado]= int (i)
                 resultado= resultado+1
             squareEach(nums)
             print ("El resultado de elevar al cuadrado es", nums)

         main()
```

```
Números al cuadrado
ingrese una lista de números separados por comas: 5,8,9
El resultado de elevar al cuadrado es [25, 64, 81]
```

1.5.5 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 [19]: def sumList (nums):
             resultado=0
             for i in nums:
                 resultado= resultado + i
             return resultado

         def main():
             print ("Sumar listas de números")
             nums = input ("Ingrese varios números separados por comas:")
             nums= nums.split (",")
             entry= 0
             for i in nums:
                 nums[entry]= int (i)
                 entry= entry + 1

             sumatotal= sumList(nums)

             print ("La suma de los números ingresados es ", sumatotal)

         main ()
```

```
Sumar listas de números
Ingrese varios números separados por comas:5,8,9
La suma de los números ingresados es  22
```

1.5.6 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 [16]: def toNumbers(strList):
             return [int(num) for num in strList]

         a= ["3","5", "7"]

         b= toNumbers (a)
         b

Out[16]: [3, 5, 7]
```