

mcpp_taller2_camila_valencia

August 17, 2016

1 Taller 2

Métodos Computacionales para Políticas Públicas - UROSARIO

Entrega: viernes 19-ago-2016 11:59 PM

[Camila Valencia] [camila.valencia@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_taller2_santiago_mataallana
- 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.

(El valor de cada ejercicio está en corchetes [] después del número de ejercicio.)

1.2 1. [1]

[Pensar como un computador] Considere el siguiente código: `if x > 2: if y > 2: z = x + y print("z es", z) else: print("x es", x)` ¿Cuál es el resultado si

- a) $x = 2, y = 5$? x es 2
- b) $x = 3, y = 1$? Nada
- c) $x = 1, y = 1$? x es 1

d) $x = 4$, $y = 3$? z es 7

1.3 2. [1][Pensar como un computador] ¿Cuál es el resultado del siguiente código y cuántas veces se recorre el loop?

`i = 0 while i < 10: i = i + 1 if i print(i)` El loop corre 10 veces

1.4 3. [1][Pensar como un computador] ¿Cuál es el resultado del siguiente código y cuántas veces se recorre el loop?

`i = 0 while i > 10: i = i + 1 if i print(i)` El loop no corre ninguna vez

1.5 4. [2]

Escriba un programa que pida al usuario ingresar un número entero, y que imprima “par” si el número es par e “impar” si el número es impar. Agregue a su programa un código que genere una advertencia en caso de que el usuario ingrese algo diferente a un número entero: “Error. El usuario debe ingresar un número entero.” (Investigue por su cuenta cómo lograr dicha validación y la generación del mensaje.)

```
In [33]: print("Ingrese un numero entero")
         x = float(input())
         if (x%2)==1:
             print(x, 'es impar')
         elif ((x+1)%2)==1:
             print(x, 'es par')
         else:
             print("Error. El usuario debe ingresar un numero entero")
```

Ingrese un numero entero

2.5

Error. El usuario debe ingresar un numero entero

1.6 5. [2]

Escriba un for loop que imprima todos los múltiplos de 3 desde 40 hasta 0 en orden decreciente. Esto es, 39, 36, 33, ..., 3, 0.

```
In [36]: for n in reversed(range(0, 40)):
         if (n%3)==0:
             print(n)
```

39

36

33

30

27
24
21
18
15
12
9
6
3
0

1.7 6. [2]

Escriba un loop que imprima todos los números entre 6 y 30 que no son divisibles por 2, 3 o 5.

```
In [35]: for n in range(6,30):  
        sirve = True  
        nums={2,3,5}  
        for i in nums:  
            if n % i == 0:  
                sirve= False  
                break  
        if sirve:  
            print(n)
```

7
11
13
17
19
23
29

1.8 7. [4]

Escriba un programa llamado “Adivine mi número”. El computador generará aleatoriamente un entero entre 1 y 100. El usuario digita un número y el computador responde “Menor” si el número aleatorio es menor que el escogido por el usuario, “Mayor” si el número aleatorio es mayor, y “¡Correcto!” si el usuario adivina el número. El jugador puede continuar ingresando números hasta que adivine correctamente.

Ejemplo: - El número aleatorio es 79. - El computador muestra el texto “Adivine el número entre 1 y 100:” y espera a que el usuario lo digite. - El usuario digita el número que está abajo en *itálicas*. - El computador devuelve uno de tres textos, según el caso: “Mayor”, “Menor”, o “¡Correcto!”.

Adivine el número entre 1 y 100: *40* Mayor

Adivine el número entre 1 y 100: 70 Mayor
Adivine el número entre 1 y 100: 80 Menor
Adivine el número entre 1 y 100: 77 Mayor
Adivine el número entre 1 y 100: 79 ¡Correcto!
¿Cómo generar números aleatorios en Python?

- Al comienzo de su programa escriba: `import random`
- Para generar un número aleatorio entre 1 y 100 escriba: `random.randint(1, 100)`

Pistas:

- Piense en qué estructuras de control le sirven para resolver el problema.
- ¿Cómo determina si el número es mayor, menor o correcto?
- ¿Cómo le da turnos adicionales al usuario para adivinar, dependiendo de si en el turno anterior adivinó o no?

```
In [6]: import random
        i=random.randint(1,100)

        veces = 0
        print("Adivina el numero entre 0 y 100")
        while veces<30:

            x=float(input())
            veces=veces+1
            if x<i:
                print("Mayor")
            if x>i:
                print("Menor")
            if x==i:
                print("¡Correcto!")
                break
```

```
Adivina el numero entre 0 y 100
55
Menor
30
Mayor
40
Mayor
45
Mayor
50
Mayor
53
Mayor
54
¡Correcto!
```