

Taller 2

Métodos Computacionales para Políticas Públicas - UROSARIO

Entrega: viernes 19-feb-2021 11:59 PM

[Juan Diego Castro Rodríguez]

[juand.castro@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_taller2_santiago_matalana
- 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. [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)
¿Cuál es el resultado si

a) x = 2, y = 5? rta: 2

b) x = 3, y = 1? rta: 3

c) x = 1, y = 1? rta: 1

d) x = 4, y = 3? rta: z es 7
```

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 % 2 == 0: print(i) respuesta: el resultado del código sería: 2,4,6 y 8. mientras que el loop se recorre 9 veces
```

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 % 2 == 0: print(i) respuesta: el resultado del código sería: 12,14,16,18,20,22... y el loop se recorre indefinidamente hasta que se interrumpa el kernel
```

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 [30]: x=input("Escriba un número:")
try:
    if int(x)%2 == 0:
        print("par")
    else:
        print("impar")
except:
    print("Error. El usuario debe ingresar un número entero")

Escriba un número:Si buenas!
Error. El usuario debe ingresar un número entero
```

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]: x=42
while x>0:
    x=x-3
    print(x)

39
36
33
30
27
24
21
18
15
12
9
6
3
0
```

6. [2]

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

```
In [74]: a=2
b=3
c=5
x=31
while x>6:
    x=x-1
    if int(x)%a!=int() and int(x)%b!=int() and int(x)%c!=int():
        print(x)

29
23
19
17
13
11
7
```

7. [4]

Escriba un programa llamado "Adivine ni 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!".

```
In [ ]: 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!</font>
```

¿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)`

```
In [112... import random
N=random.randint(1,100)
x=0
while int(x)!=int(N):
    x=input("Adivine el número entre 1 y 100: ")
    if int(x)>int(N):
        print("menor")

    elif int(x)<int(N):
        print("mayor")
    print()
print("correcto")
```

Adivine el número entre 1 y 100: 50
menor
Adivine el número entre 1 y 100: 40
menor
Adivine el número entre 1 y 100: 30
mayor

Adivine el número entre 1 y 100: 35
mayor

Adivine el número entre 1 y 100: 36
mayor

Adivine el número entre 1 y 100: 37
mayor

Adivine el número entre 1 y 100: 38
correcto

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?