

Taller 2

Métodos Computacionales para Políticas Públicas - URosario

Entrega: viernes 14-feb-2020 11:59 PM

**Santiago Ortiz Ortiz **

santiago.ortizo@urosario.edu.co (<mailto:santiago.ortizo@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_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. [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) $x = 2$, $y = 5$?

b) $x = 3$, $y = 1$?

c) $x = 1$, $y = 1$?

d) $x = 4$, $y = 3$?

Respuesta

a) Como x es igual a 2, el primer "if" no lo va a tomar, por lo tanto pasamos de una vez al "else". Por lo tanto, la condición que va a cumplir será `print(x)`. Es decir, el resultado será 2.

b) Como x es igual a 3, el primer "if" lo va a tomar. Sin embargo, no hará nada. Como $y = 1$, no va a hacer nada en esa parte. Por lo tanto el retorno de ese código no será nada. Solo correrá sin errores.

c) Dado que x es 1, el primer condicional no se activará, por lo tanto el resultado es `print(x)`, es decir, 1.

d) Dado que se cumple el primer condicional, y adicional a eso, se cumple el condicional que tiene dentro, el print será: 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

Como i comienza en 0, el while loop comenzará, lo primero que hace es ir aumentando el valor de i . Como la condición "if" para imprimir es que el número sea par, el resultado del código será: 2, 4, 6, 8, 10. El loop se recorre 10 veces. La primera cuando i es 0 hasta cuando i es 9. El resultado es un i 10, pero porque en el loop de i 9, aumentan 1. El loop para cuando i es 10, porque el while así lo determina.

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 loop nunca se ejecutará porque i es 0, lo cual es menor a 10. No hay resultado.

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 [5]: booleano = True
while booleano == True:
    try:
        numero = float(input("Por favor ingrese un número entero: "))
        while numero != int(numero):
            print("Error. El usuario debe ingresar un número entero")
            numero = float(input("Por favor ingrese un número entero: "))
        if numero == int(numero):
            if numero % 2 == 0:
                print(f"{numero} es par")
                booleano = False
            else:
                print(f"{numero} es impar")
                booleano = False
        except ValueError:
            print("Error. El usuario debe ingresar un número entero.")
```

```
Por favor ingrese un número entero: hola
Error. El usuario debe ingresar un número entero.
Por favor ingrese un número entero: si
Error. El usuario debe ingresar un número entero.
Por favor ingrese un número entero: 6.5
Error. El usuario debe ingresar un número entero
Por favor ingrese un número entero: 5
5.0 es impar
```

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 [43]: for i in range(40,-1,-1):  
         if i % 3 ==0:  
             print(i)
```

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

```
In [20]: for i in range(41):  
         i=40-i  
         if i % 3 == 0:  
             print(i)
```

```
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 [3]: divisores = [2,3,5]
        for j in range(6,31):
            for i in divisores:
                if j % i != 0:
                    print(f"{j} NO divisible por {i}")
```

```
6 NO divisible por 5
7 NO divisible por 2
7 NO divisible por 3
7 NO divisible por 5
8 NO divisible por 3
8 NO divisible por 5
9 NO divisible por 2
9 NO divisible por 5
10 NO divisible por 3
11 NO divisible por 2
11 NO divisible por 3
11 NO divisible por 5
12 NO divisible por 5
13 NO divisible por 2
13 NO divisible por 3
13 NO divisible por 5
14 NO divisible por 3
14 NO divisible por 5
15 NO divisible por 2
16 NO divisible por 3
16 NO divisible por 5
17 NO divisible por 2
17 NO divisible por 3
17 NO divisible por 5
18 NO divisible por 5
19 NO divisible por 2
19 NO divisible por 3
19 NO divisible por 5
20 NO divisible por 3
21 NO divisible por 2
21 NO divisible por 5
22 NO divisible por 3
22 NO divisible por 5
23 NO divisible por 2
23 NO divisible por 3
23 NO divisible por 5
24 NO divisible por 5
25 NO divisible por 2
25 NO divisible por 3
26 NO divisible por 3
26 NO divisible por 5
27 NO divisible por 2
27 NO divisible por 5
28 NO divisible por 3
28 NO divisible por 5
29 NO divisible por 2
29 NO divisible por 3
29 NO divisible por 5
```

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!".

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!

```
In [18]: import random
aleatorio = (random.randint(1,100))
numero = int(input("Adivine el número entre 1 y 100: "))
while aleatorio != numero:
    if aleatorio > numero:
        print("Mayor")
        numero = int(input("Adivine el número entre 1 y 100: "))
    if aleatorio < numero:
        print("Menor")
        numero = int(input("Adivine el número entre 1 y 100: "))
if numero == aleatorio:
    print("¡Correcto!")
```

```
Adivine el número entre 1 y 100: 7
Mayor
Adivine el número entre 1 y 100: 90
Menor
Adivine el número entre 1 y 100: 50
Mayor
Adivine el número entre 1 y 100: 60
Mayor
Adivine el número entre 1 y 100: 70
Mayor
Adivine el número entre 1 y 100: 80
Mayor
Adivine el número entre 1 y 100: 90
Menor
Adivine el número entre 1 y 100: 87
Mayor
Adivine el número entre 1 y 100: 88
¡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?

