

# Taller 2

Métodos Computacionales para Políticas Públicas - UROSARIO

Entrega: viernes 28-ago-2020 11:59 PM

**\*\*[Nicolás Garcés Rodríguez]\*\***

[nicolas.garces@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?

**a) x = 2, y = 5?**

x no es >2, entonces el if no entrará en acción. Por ende solo se imprimirá x. El resultado será 2

In [ ]:

```
x=2
y= 5
if x > 2:
    if y > 2:
        z = x + y
        print("z es", z)
else:
    print(x)
```

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

x si es > 2, pero Y no cumple su condicional (1 no es >3), como no hay otra condición para cuándo y sea menor o igual a 2 el código no tendrá ningún output.

In [ ]:

```
x=3
y=1
if x > 2:
    if y > 2:
        z = x + y
        print("z es", z)
else:
    print(x)
```

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

x no cumple la primera condición, entonces solo se imprimirá x. el resultado es 1.

In [ ]:

```
x=1
y=1
if x > 2:
    if y > 2:
        z = x + y
        print("z es", z)
else:
    print(x)
```

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

tanto x como y cumplen con las condiciones de sus if, por ende el output será: "z es 7", donde z es la suma de x y Y.

In [ ]:

```
x=4
y=3
if x > 2:
    if y > 2:
        z = x + y
        print("z es", z)
else:
    print(x)
```

---

## 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)
```

El código tendrá como output los números 2,4,6,8,10. El loop se correrá 10 veces. cada vez que i pase por el loop ira creciendo de a una unidad, pero la condición para imprimir solo se cumplirá cada dos loop (cuando i sea par), sin contar el loop inicial, por ende solo generará un output cuando este en un número de repeticiones impar.

---

## 3. [1]

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

[Pensar como un computador] ¿Cual es el resultado del siguiente código y cuantas veces se recorre el loop?

```
i = 0 while i > 10: i = i + 1 if i % 2 == 0: print(i)
```

El código no tendrá ningún output. Como el loop while corre solo cuando  $i > 10$ . Como  $i$  no cumple esa condición, no pasará nada.

## 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 [2]:

```
num = float(input("ingrese un numero entero:"))
while num % 1 > 0:
    print("Error. El usuario debe ingresar un número entero.")
    num = float(input("ingrese un numero entero:"))
x = int(num)
if x % 2 == 0:
    print(f"{x} es par")
else:
    print(f"{x} es impar")
```

```
ingrese un numero entero:4.5
Error. El usuario debe ingresar un número entero.
ingrese un numero entero:3.2
Error. El usuario debe ingresar un número entero.
ingrese un numero entero:7.6
Error. El usuario debe ingresar un número entero.
ingrese un numero entero:4
4 es par
```

## 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 [3]:

```
for i in reversed(range(0,40)):
    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 [4]:

```
for i in range(6,30):  
    if i%2 != 0 and i%3 != 0 and i%5 != 0:  
        print(i)
```

```
7  
11  
13  
17  
19  
23  
29
```

## 7. [4]

Escriba un programa llamado "Adivine el 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!</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)`

1. Ideas.

- 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 [1]:

```
##### un numero aleatorio para todas las rondas

import random
print("Adivine el número")
ran = random.randint(1, 100)
num = int(input("Adivine el número entre 1 y 100:"))
while num != ran:
    if num > ran:
        print("Mayor")
    else:
        print("Menor")
    num = int(input("Adivine el número entre 1 y 100:"))
if num == ran:
    print(";Correcto!")
```

```
Adivine el número
Adivine el número entre 1 y 100:50
Mayor
Adivine el número entre 1 y 100:30
Menor
Adivine el número entre 1 y 100:32
Menor
Adivine el número entre 1 y 100:40
Menor
Adivine el número entre 1 y 100:45
Mayor
Adivine el número entre 1 y 100:42
Menor
Adivine el número entre 1 y 100:43
;Correcto!
```

In [ ]:

```
##### un numero aleatorio cada ronda

import random
print("Adivine el número")
ran = random.randint(1, 100)
num = int(input("Adivine el número entre 1 y 100:"))
while num != ran:
    if num > ran:
        print("Mayor")
    else:
        print("Menor")
    ran = random.randint(1, 100)
    num = int(input("Adivine el nuevo número entre 1 y 100:"))

if num == ran:
    print(";Correcto!")
```

```
Adivine el número
Adivine el número entre 1 y 100:50
Menor
Adivine el nuevo número entre 1 y 100:50
Mayor
```

---