

# Taller 2

Métodos Computacionales para Políticas Públicas - URosario

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

Juan Sebastián Gómez

Juansebastian.gomez@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\_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.

(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). El programa evalúa si  $x$  es mayor a 2, al ver que no es mayor deja de ejecutar las siguientes líneas de código indentadas y pasa a ejecutar el condicional `else`, por lo que el resultado es 2.

b). Dado que  $x$  es mayor a 2 se cumple el primer condicional, pero, el programa, al evaluar el siguiente condicional ve que  $y$  es menor que 2 y no se cumple, por lo que no arroja ningún resultado.

c). El programa evalúa si  $x$  es mayor a 2, al ver que no es mayor ignora las siguientes líneas de código indentadas (no las ejecuta) y pasa a ejecutar el condicional `else`, por lo que el resultado es 1.

d). Dado que  $x$  es mayor a 2 se cumple el primer condicional, y se procede a evaluar el segundo condicional, donde también se cumple dado que  $y$  es mayor a 2, por lo que el programa ejecuta la operación  $z = x + y$  e imprime como resultado:  $z$  es 7. No imprime ( $x$ ), porque al cumplirse los condicionales anteriores, el programa no evalua la línea de código del condicional "else".

---

## 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 loop corre 10 veces, cada vez se la va a sumar 1 a la variable ( $i$ ), y se pide que imprima ( $i$ ) solo si el residuo de ( $i$ ) al dividirle por 2 es igual a 0, por lo que el resultado es:

- 2
  - 4
  - 6
  - 8
  - 10
-

### 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)
```

El programa va a evaluar si se cumple la condición de que (i) es > 10 para continuar con la ejecución de las operaciones, pero como no se cumple la condición, el programa no arroja resultado y se detiene después de la primera evaluación de la condición.

---

### 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.)

---

1. Programa pide ingresar un número y evalúa si es par o impar. Si el unuario no ingresa un número entero le arroja un mensaje de error

In [2]:

```
try:  
    num = int(input("Digite un número: " ))  
    if num % 2 == 0:  
        print(num, "es par")  
    else:  
        print(num, "es impar")  
  
except:  
    print("Error. El usuario debe ingresar un número entero")
```

Error. El usuario debe ingresar un número entero

In [3]:

```
try:
    num = int(input("Digite un número: " ))
    if num % 2 == 0:
        print(num, "es par")
    else:
        print(num, "es impar")

except:
    print("Error. El usuario debe ingresar un número entero")
```

5 es impar

In [4]:

```
try:
    num = int(input("Digite un número: " ))
    if num % 2 == 0:
        print(num, "es par")
    else:
        print(num, "es impar")

except:
    print("Error. El usuario debe ingresar un número entero")
```

4 es par

1. Programa pide ingresar un número y evalúa si es par o impar. Si el unuario no ingresa un número entero le arroja un mensaje de error y le permite ingresar otro número

In [5]:

```
while True:
    try:
        num = int(input("Digite un número: " ))
    except ValueError:
        print("Error. El usuario debe ingresar un número entero")
        continue
    else:
        break
    if num % 2 == 0:
        print(num, "es par")
    else:
        print(num, "es impar")
```

Error. El usuario debe ingresar un número entero

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 [6]:

```
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 [7]:

```
for i in range (6,30):
    if (i % 2 != 0) :
        print (f"{i} no es divisible por 2")
    if (i % 3 != 0) :
        print (f"{i} no es divisible por 3")
    if (i % 5 != 0) :
        print (f"{i} no es divisible por 5")
```

```
6 no es divisible por 5
7 no es divisible por 2
7 no es divisible por 3
7 no es divisible por 5
8 no es divisible por 3
8 no es divisible por 5
9 no es divisible por 2
9 no es divisible por 5
10 no es divisible por 3
11 no es divisible por 2
11 no es divisible por 3
11 no es divisible por 5
12 no es divisible por 5
13 no es divisible por 2
13 no es divisible por 3
13 no es divisible por 5
14 no es divisible por 3
14 no es divisible por 5
15 no es divisible por 2
16 no es divisible por 3
16 no es divisible por 5
17 no es divisible por 2
17 no es divisible por 3
17 no es divisible por 5
18 no es divisible por 5
19 no es divisible por 2
19 no es divisible por 3
19 no es divisible por 5
20 no es divisible por 3
21 no es divisible por 2
21 no es divisible por 5
22 no es divisible por 3
22 no es divisible por 5
23 no es divisible por 2
23 no es divisible por 3
23 no es divisible por 5
24 no es divisible por 5
25 no es divisible por 2
25 no es divisible por 3
26 no es divisible por 3
26 no es divisible por 5
27 no es divisible por 2
27 no es divisible por 5
28 no es divisible por 3
28 no es divisible por 5
29 no es divisible por 2
29 no es divisible por 3
29 no es 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 *italicas*.
- 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)`

### 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 [8]:

```
import random
```

In [9]:

```
print("Juego adivine mi número")
name_u = input("Ingrese un nombre de usuario: ")
print(f"Hola {name_u}, voy a pensar en un número entre el 1 y 100, el juego termina cuando adivines el número")
my_num = random.randint(1,100)
while True:
    try:
        user_num = int(input("Adivina el número que he pensado: "))
    except ValueError:
        print("Solo se permiten números enteros")
        continue
    if user_num < 0 or user_num > 100:
        print("Solo son números entre 1 y 100, inténtalo de nuevo")
        continue
    else:
        break
while user_num != my_num:
    if user_num < my_num :
        print("Mayor")
    else :
        print ("menor")
    user_num = int(input("Adivina el número que he pensado: "))
if user_num == my_num :
    print ("¡Correcto!")
```

Juego adivine mi número

Hola Juan, voy a pensar en un número entre el 1 y 100, el juego termina cuando adivines el número

Solo se permiten números enteros

Solo son números entre 1 y 100, inténtalo de nuevo

menor

Mayor

menor

menor

menor

Mayor

¡Correcto!