Taller 2

Métodos Computacionales para Políticas Públicas - URosario

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

[Monica Alejandra Robayo Gonzalez]

[monica.robayo@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) x = 2, y= 5?

In [62]:

```
x = 2
y = 5

if x > 2: #la condicion no se cumple
    if y > 2: # la condicion se cumple
        z = x + y # como la condicion x > 2 no se cumple, el programa toma como salida
 else y no realiza la suma.
        print("z es", z)
else:
    print(x) # el programa imprime x, toda vez que la condicion de arriba no se cumple.
El resultado seria 2
```

2

b) x = 3, y = 1?

In [58]:

```
x = 3
y = 1

if x > 2: #la condicion se cumple
    if y > 2:# la condicion no se cumple
        z = x + y
        print("z es", z)
else:
    print(x) #el programa no ejecuta este else, toda vez que corresponde al primer if x
>2
```

In [65]:

```
#si queremos ejecutar print (x) del segundo if,ponemos un else dentro del segundo if, a
si:
x = 3
y = 1

if x > 2: #la condicion se cumple
    if y > 2: # la condicion no se cumple
        z = x + y
        print("z es", z)
    else:
        print(x) # como la segunda condicion no se cumple, el programa ejecuta x, lo qu
e seria igual a 3.
```

3

c) x = 1, y = 1?

In [67]:

```
x = 1
y = 1

if x > 2: #la condicion no se cumple
    if y > 2: #la condicion no se cumple
        z = x + y #ninguna de las dos condiciones son ciertas, el programa toma como sa
lida else y no realiza la suma.
        print("z es", z)
else:
    print(x)  # el programa imprime x, toda vez que las condiciones de arriba no se cum
plen. El resultado seria 1
```

```
1
```

d) x = 4, y = 3?

In [41]:

```
x = 4
y = 3

if x > 2: #la condicion se cumple
    if y > 2: #la condicion se cumple
        z = x + y # como las dos condiciones se cumplen, el programa realiza la suma x
 + y
        print("z es", z) #el resultado es 7
else:
    print(x) # el programa no ejecuta el else, toda vez que las condiciones de arriba s
on ciertas
```

```
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)

In [48]:

```
i = 0 #el programa ejecuta la primera instrucción donde i = 0
while i < 10: #se ejecuta la condicion del while, como la condición se cumple,el progra
ma ejecuta la instrucción donde i<10
    i = i + 1 #enseguida lo que hace el programa es ejecutar los otros dos bloques dond
e i=i+1
    if i % 2 == 0:# entonces si el residuo de i%2 es cero, el programa imprime todos lo
s i donde el residuo sea cero.
        print(i)
```

```
2
4
6
8
10
```

Paso a Paso

In [4]:

```
i = 0 # primera instrucción i=0
```

In [56]:

```
i < 10 # la condición es que i sea menor que 10, como se cumple el programa ejecuta los
siguientes dos bloques
```

Out[56]:

```
True
```

In [57]:

```
i = i + 1 # se modifica la variable i=0 por i = i+1
```

In [55]:

```
#el programa ejecuta la operación donde i = i+1 teniendo en cuenta que i<10, toma unica
mente los numeros de 0 a 9.
print (0 + 1)
print (1 + 1)
print (2 + 1)
print (3 + 1)
print (4 + 1)
print (5 + 1)
print (6 + 1)
print (7 + 1)
print (8 + 1)
print (9 + 1)
```

```
1
2
3
4
5
6
7
8
9
10
```

In [59]:

```
# una vez el programa tenga los valores de i, realiza la siguiente instrucción i%2
print (1 % 2)
print (2 % 2)
print (3 % 2)
print (4 % 2)
print (5 % 2)
print (6 % 2)
print (7 % 2)
print (8 % 2)
print (9 % 2)
print (10 % 2)
```

```
1
0
1
0
1
0
1
0
1
0
```

In [62]:

```
# sin embargo como la instrucción incluye que el residuo del cociente sea igual a cero, el programa solo toma aquella division que cumple con la condicion, asi:
print (2 % 2)
print (4 % 2)
print (6 % 2)
print (8 % 2)
print (10 % 2)
```

```
0
0
0
0
0
```

In [63]:

```
# por ultimo la instrucción indica que el programa solo debe imprimir aquellos resultados de i, donde el residuo de i%2 sea cero, por tanto solo imprime los siguientes resultados:
print (1 + 1)
print (3 + 1)
print (5 + 1)
print (7 + 1)
print (9 + 1)
```

```
2
4
6
8
10
```

In [ ]:

```
# como resultado del codigo tenemos que es : 2,4,6,8,10 y el loop se estaria corriendo
 10 veces (desde cero hasta 9), y solo imprime lo que dice la instrucción (residuos iguales a cero)
```

---

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)

In [64]:

```
i = 0 # la instrucción es que la  variable i es igual a cero (0)
while i > 10: # se comprueba si la variable (i > 10), como la condicion no se cumple, los dos bloques siguientes no se ejecutan.
    i = i + 1
    if i % 2 == 0:
        print(i)
```

In [ ]:

```
# Respuesta: el código no se ejecuta toda vez que la condición no se cumple, por tanto
 el loop no es ejecutado.
```

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 [ ]:

```
# para este punto consulte las excepciones en python, lo que se llama "try" y "except",
los cuales sirven para ejecutar en bloque el codigo, esto es: si se cumple lo que esta
 dentro del "try" es decir la declaracion, el programa ejecuta el "try", de lo contrari
o aplica una excepcion.
```

In [3]:

```
print ("par e impar")

try:
    print ("digame un numero entero")
    num = int(input())
    if type (num) == int:
        if num % 2 == 0:
            print("el numero es par")

        else:
            print("el numero es impar")

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

```
par e impar
digame un numero entero
10
el numero es par
```

In [4]:

```
print ("par e impar")

try:
    print ("digame un numero entero")
    num = int(input())
    if type (num) == int:
        if num % 2 == 0:
            print("el numero es par")

        else:
            print("el numero es impar")

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

```
par e impar
digame un numero entero
9
el numero es impar
```

In [5]:

```
print ("par e impar")

try:
    print ("digame un numero entero")
    num = int(input())
    if type (num) == int:
        if num % 2 == 0:
            print("el numero es par")

        else:
            print("el numero es impar")

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

```
par e impar
digame un numero entero
sdf
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 [29]:

```
for multiplo in range (40,-1,-1):
    if (multiplo % 3) == 0:
        print (multiplo)
```

```
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 o 3 o 5.

In [2]:

```
for x in range(6,31):
    if (x % 2) !=0:
        print (x, "No es divisible por 2")
    if (x % 3) !=0:
        print (x, "No es divisible por 3")
    if (x % 5) !=0:
        print (x, "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 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)`

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 [1]:

```
print ("Adivine mi número")
import random
y = random.randint(1, 100)

x = int(input())

while x!=y:

 if x<y:
        print (x,"Menor")
        x=int (input("Ingresa un numero entre 1 y 100: "))

 elif x>y:
        print (x,"Mayor")
        x=int (input("Ingresa un numero entre 1 y 100: "))
else:
    print(x,"¡Correcto!")
```

```
Adivine mi número
50
50 Mayor
Ingresa un numero entre 1 y 100: 40
40 Mayor
Ingresa un numero entre 1 y 100: 10
10 Menor
Ingresa un numero entre 1 y 100: 20
20 Mayor
Ingresa un numero entre 1 y 100: 15
15 ¡Correcto!
```