Taller 2

Métodos Computacionales para Políticas Públicas - URosario

Entrega: viernes 15-feb-2019 11:59 PM

César Mateo Lancheros Cañóm

cesar.lancheros@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 es", 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?

Respuestas:

a. x es 2: No se cumple el primer condicional, así que salta y se ejecuta el código print final.

b. Sin resultado: Se cumple el primer condicional pero no el segundo, así que no existe ninguna opción.

c. x es 1: No se cumple ni el primer ni el segundo condicional, por lo que salta y se ejecuta el código print final.

d. z es 7: Se cumplen ambos condicionales por lo que se se ejecuta la secuencia hasta la operación.

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)

Los resultados son:

2; 4; 6; 8; 10

El código se recorre 10 veces, pero arroja solamente cinco valores. Dado que, el código toma los número enteros entre 0 y 10, pero unicamente aquellos que al tomar el valor adicional de 1 son divisibles entre 2. Por esta razón, el resultado son todos los números pares entre dicho intervalo.

---

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)

Sin resultado

Según el código la primera línea anuncia que i=0, mientras que en la segunda se invalida la acción dado que i no es mayor que cero. Por este motivo, el código no puedo continuar ejecutandose y no se arroja 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 [ ]:
```
while True:
    value = input("Ingresar un número entero: ")
    try:
        value = int(value)
        if value % 2 == 0:
            print ("par")
        else:
            print ("impar")


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

return valor
```

```
Ingresar un número entero: 75
impar
Ingresar un número entero: 88
par
Ingresar un número entero: 45.7
Error. El usuario debe ingresar un número entero.
Ingresar un número entero: 36.8
Error. El usuario debe ingresar un número entero.
Ingresar un número entero: 100
par
Ingresar un número entero: 155
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 [19]:
```
i = [40,39,38,37,36,35,34,33,32,31,30,29,28,27,26,25,24,23,22,21,19,18,
17,16,15,14,13,12,11,10,9,8,7,6,5,4,3,2,1,0]
for x in i:
    if (x % 3) == 0:
        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 [27]:
```
j = range(6,30)
```

```
for y in j:
    if (y % 2) != 0 and (y % 3) != 0 and (y % 5) != 0:
        print(y)
```

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

---

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 [2]:
```
import random
int = 0
z = random.randint(1,100)

while int < 100:
    k = eval (input("Adivine el número entre 1 y 100: "))
    if k < z:
        print("Mayor")
    if k > z:
        print("Menor")
    if k == z:
        break

if k == z:
    print ("¡Correcto!")
```

```
Adivine el número entre 1 y 100: 69
Mayor
Adivine el número entre 1 y 100: 84
Menor
Adivine el número entre 1 y 100: 72
Mayor
Adivine el número entre 1 y 100: 76
Menor
Adivine el número entre 1 y 100: 74
¡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?
