

mcpp_taller2_andrés_ramírez

February 14, 2020

1 Taller 2

Métodos Computacionales para Políticas Públicas - URosario

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

[Andrés Ramírez] [andrese.ramirez@urosario.edu.co]

1.1 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.)

1.2

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

[]: Solución

- a) La respuesta es 2, debido a que x no es mayor a 2, por ende, imprimirá el valor de x
 - b) El código no tendría sentido para la maquina, debido al hecho de que aunque se cumple el condicional para x , no es así para el caso de y . Además, como no se pone un `else` al `if` que condiciona el y , la maquina no tendría como dar un resultado.
 - c) En este caso la maquina arrojaría como resultado el 1, por cuanto el valor de x no es mayor a 2.
 - d) En este caso el valor sería 7, pues se cumplirían con las condiciones para hallar z .
-

1.3

[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 print(i)` Solución

En este caso el loop se recorre cinco veces y daría como resultado los números pares menores o iguales a 10. Esto es, 2,4,6,8 y 10.

1.4

[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 print(i)` Solución

En este caso la maquina no podría correr el código o arrojar un resultado. Lo anterior, por cuanto i es igual a cero, entonces, si solo se puede tomar los valores cuando i es mayor a 10, pues no habría manera de hallar un número que haga verdad la expresión que acompaña al `while`.

[]:

1.5

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.)

Solución

```
[6]: a = float(input("Por favor escriba un número entero: "))

b = int (a)

while not (a==b):
    print("Error. El usuario debe ingresar un número entero.")
    a = float(input("Por favor escriba un número entero: "))
if (a == b):
    if (b%2 == 0) and (a == b):
        print(f"{b} es un número par.")
    if not (b%2 == 0) and (a == b):
        print(f"{b} es un número impar.")
```

```
Por favor escriba un número entero: 3.1415
Error. El usuario debe ingresar un número entero.
Por favor escriba un número entero: 3.244556
Error. El usuario debe ingresar un número entero.
Por favor escriba un número entero: 6.578564
Error. El usuario debe ingresar un número entero.
Por favor escriba un número entero: 12354.6889
Error. El usuario debe ingresar un número entero.
Por favor escriba un número entero: 3
3 es un número impar.
```

1.6

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.

Solución

```
[9]: for i in range(41,0-1,-1):  
      if (i%3)==0 ):  
          print(i)  
  
print("")
```

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

1.7

Escriba un loop que imprima todos los números entre 6 y 30 que no son divisibles por 2, 3 o 5.

```
[20]: x = 2
      y = 3
      z = 5

      for i in range(6,31):
          if not ((i%x==0)):
              print(f"{i} no es divisible por {x}")
          if not ((i%y ==0)):
              print(f"{i} no es divisible por {y}")
          if not (i%z==0):
              print(f"{i} no es divisible por {z}")

      print("")
```

```
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

1.8

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!

¿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?

Solución

```
[ ]: import random
número = random.randint(1, 100)

encontrado = False
turno = 0

while not encontrado:

    persona = int(input("Introduzca un número: "))

    if (persona > número):
        print("El número es menor")
        turno = turno + 1
    elif persona < número:
        print("El número es mayor")
        turno = turno + 1
    else:
        encontrado = True
        print(f" {persona} es el número correcto")
```