

mcpp_taller2_john_caro

August 17, 2016

1 Taller 2

Métodos Computacionales para Políticas Públicas - UROSARIO

Entrega: viernes 19-ago-2016 11:59 PM

[John Alexander Caro Becerra] [jhonalexbc@gmail.com]

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 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) $x = 4, y = 3$?
- a) el resultado de la programación es "x es 2"
- b) el resultado de la programación es vacío, o sea no sale nada
- c) el resultado de la programación es "x es 1"
- d) el resultado de la programación es "z es 7"

1.3 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 print(i) El resultado del código es una secuencia de números pares comenzando desde 0 hasta 10 es decir

0,2,4,6,8,10

Pero las veces que se corre son 10 veces

1.4 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 print(i) no da resultado ya que como $i = 0$, no realiza el loop ni una vez y de paso no se corre el loop

1.5 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]: print("Por favor ingrese un número entero")
        a = input()
        try:
            aa = int(a)
            if type(aa) != int:
                print("Error. El usuario debe ingresar un número entero.")
            if (aa % 2) == 0:
                print("El número ingresado:", aa, " es par")
            else:
                print("El número ingresado:", aa, " es impar")
        except ValueError:
            print("Error. El usuario debe ingresar un número entero, o un número al")
            pass
```

Por favor ingrese un número entero

5

El número ingresado: 5 es impar

1.6 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 [24]: #len(range(0,41))
x = 41
for i in range(41):
    x = x - 1
    if (x % 3) == 0:
        if (x != 0):
            print(x)
```

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

1.7 6. [2]

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

```
In [38]: for j in range(6,31):
z = j
if (z % 2) != 0:
    if (z % 3) != 0:
        if (z % 5) != 0:
            print(z)
```

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

1.8 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!

¿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 [3]: `import random`

```
num_random = random.randint(1, 100)
print("Adivine el número entre 1 y 100: Ingrese un número")
b = input()
try:
    aa = int(b)
    if type(aa) != int:
        print("Error. El usuario debe ingresar un número entre 1 y 100.")
except ValueError:
    print("Error. El usuario debe ingresar un número entre 1 y 100.")
    b = input()
    aa = int(b)
    pass
for z in range(1000000):
    if aa < num_random:
        print("Mayor")
        b = input()
        aa = int(b)
    elif aa > num_random:
        print("Menor")
```

```
        b = input()
        aa= int(b)
    else:
        print("¡Correcto!")
        break
```

Adivine el número entre 1 y 100: Ingrese un número

45

Menor

40

Menor

30

Menor

20

Mayor

25

Menor

24

Menor

23

Menor

22

¡Correcto!