

Escalona_Joaquin_2

August 22, 2018

Hola, pueden encontrar este documento y los códigos aquí copiados para que puedan ejecutarlos y evaluarlos con mayor facilidad pinchando aquí :) [GitHub](#)

1 Ejercicio 1

- A) Usen "time" para evaluar cuanto tiempo se demora en correr este programa con un numero muy grande para "max". Que valor de "max" tienen que adoptar para que pasa un tiempo humanamente perceptible?

```
max = int(raw_input(Inserta un numero entero positivo: ))
i=0
while i < max:
    i = i + 1
    print i
```

- B) Quiten el "print" dentro del loop. Cuanto tiempo se demora correr este programa con el mismo "max"?

```
In [ ]: #PARA CALCULAR EL TIEMPO DE DEMORA
from time import time
# UN NUMERO MUY GRANDE ES, POR EJEMPLO, 2 MILLONES
max=int(raw_input('Inserta un numero entero positivo: '))
#ACA VA EL INICIAL TIME YA QUE EL USUARIO SE PUEDE DEMORAR EN INGRESAR EL NUMERO
inicial=time()
i=0
#CICLO QUE FINALIZA CUANDO I > MAX
while i < max:
    i = i+1
    print i
final = time()
#TIEMPO TOTAL
print(final-inicial)
#SI EL USUARIO INGRESA 2 MILLONES, EL PROGRAMA TARDA 9.48944306374(s)
#ESTE TIEMPO CAMBIA EN CADA INTENTO, PERO APROXIMADAMENTE SON 9(s)
```

Solución al apartado A) : Considero que un humano se dará cuenta si transcurre un lapso de 2 segundos. Así que un tiempo de 2(s) es suficiente. La solución a continuación dará el máx cuando el programa lleve un poco más de 2 segundos corriendo :)

```
In [ ]: # PARA PARAR EL PROGRAMA AL CABO DE X SEGUNDOS IMPORTAMOS CLOCK
# IDEA SACADA DE: # https://recursospython.com/guias-y-manuales/scheduler/
from time import clock
#TIEMPO INICIAL
inicial=0
i=0
while True:
#TIEMPO ACTUAL
    actual=clock()
    i = i+1
    print i
#SI EL TIEMPO ES MAYOR A 2 SEGUNDOS, PARAR
    if (actual - inicial) > 2:
        break
```

El programa anterior arroja distintos valores cada vez que se ejecuta, entre ellos

- 392206
- 396469
- 392972
- 388306

Vemos que todos ellos son cercanos, así que elegimos por ejemplo, al **392206** como nuestro valor max para que pase un tiempo perceptiblemente humano. Al evaluar este numero en el programa anterior (propuesto por la profesora), el tiempo que arroja aproximadamente es de 2(s) (funciona :D !!)

Solucion al apartado B): Al eliminar el print, el programa arroja para 392206 un tiempo de 0.051815032959 (s) aprox. Mucho menor que 2 (s)

2 Ejercicio 2

Escriban un programa usando “for loops” que identifique cuales de los números entre 10 y 19 son números primos.

```
In [ ]: #DEFINIMOS FUNCION QUE ARROJA SI UN VALOR ES PRIMO O NO
#INPUT = X
#SI X ES PRIMO, RETORNA 1
#SI NO, RETORNA 0

def es_primo(x):
#INICIO LOOP DESDE 2 HASTA X-1
    for i in range(2,x):

#LA FUNCION -IF ANY(LISTA)- LA APRENDI EN SOLOLEARN (SECCION "MAS TIPOS >> FUNCIONES
#UTILES")
#PARA QUE UN NUMERO SEA PRIMO, ESTE DEBE SER DIVISIBLE SOLO POR 1 Y EL MISMO
#POR LO TANTO SI EXISTIERA ALGUN i (IF ANY) QUE EL CUOCIENTE ENTRE EL NUMERO E i
#SEA IGUAL A 0
```

```

#EL NUMERO NO ES PRIMO (Y LA FUNCION RETORNARA 0)
    if any([x%i==0]):
        return 0
    return 1

#LISTA DE PRIMOS INICIALMENTE VACIA
primos=[]

#PROBAR ENTRE 10-19
for j in range(10,20):
    if es_primo(j)==1:
        primos.append(j)

print (' Lista de primos entre 10 y 19 es \n '+str(primos))

```

El programa dice que [11, 13, 17 y 19] son primos.

3 Ejercicio 3

Let s be a string that contains a sequence of decimal numbers separated by commas, e.g., s = '1.23,2.4,3.123'. Write a program that prints the sum of the numbers in s .

Link de donde saqué la idea de usar función `eval`: [eval\(\)](#)

```

In [ ]: #EJEMPLO CADENA CON DECIMALES
s = "1.23,2.4,3.123"

#SE USA LA FUNCION EVAL() PARA CREAR UNA LISTA CON LOS NUMEROS QUE TENGA LA CADENA
#ANTERIOR
#EL LINK DONDE SAQUE LA IDEA ES CITADO EN EL ENUNCIADO
l=eval(s)
#PARA SUMAR TODOS LOS ELEMENTOS EN UNA LISTA, SE USA SUM()
print 'La suma de todos los numeros es', sum(l)

```

4 Ejercicio 4

Usando “break” modifiquen el siguiente programa para que salga del loop cuando i = 7.

```

import time
i=1
while True:
    print "Welcome", i, "times. To stop press [CTRL+C]"
    i += 1
    #Delay for 2 seconds
    time.sleep(2)

In [ ]: import time

```

```
i=1
while True:
    print "Welcome", i, "times. To stop press [CTRL + C]"
    i+=1
    #Delay for 2 seconds
    time.sleep(2)
    #SI I ES IGUAL A 7, BREAK. (EL PROGRAMA SALUDARA 6 VECES )
    if i == 7:
        break
```