

# Escalona\_Joaquin\_1

August 11, 2018

Hola, pueden encontrar este documento en su version de notebook, además de todos los programas aquí copiados para que puedan ejecutarlo y evaluarlo con mayor facilidad pinchando [GitHub](#) :)

## 1 Ejercicio 1:

Which of the following is a Boolean expression? Select all that apply:

- (A) True
  - (B) 3==4
  - (C) 3+4
  - (D) 3+4 == 7
- 

## 2 Ejercicio 2:

Boolean operators: Returned values in logical test, the returned value is the one that has been evaluated last. Consider these examples:

```
In [1]: print (True and 'Ok' or 'Ko')
```

Ok

```
In [2]: print(False and 'Ok' or 'Ko')
```

Ko

Explanation: in the first statement, True and "OK" is True. There is not need to test or "KO", so this is the end of the logical test, and the returned value is the one that has been evaluated last in True and "OK". In the second statement, False and "OK" is False. So, or "KO" must be evaluated. So, the last evaluated expression is "KO", hence the returned value.

Evalúen las siguientes expresiones, y expliquen el resultado:

```
In [3]: 'a' == ('a' or 'b')
```

Out[3]: True

Es *True* debido a que el carácter "a" fue el ultimo valor retornado y entonces es cierto que a = a

```
In [4]: 'b' == ('a' or 'b')
```

```
Out[4]: False
```

Es *Falso* debido a que "a" fue el ultimo valor retornado y es falso que 'b'='a'

```
In [5]: 'a' == ('a' and 'b')
```

```
Out[5]: False
```

Es *Falso* debido a que "b" fue el ultimo valor retornado y es falso que 'a'='b'

```
In [6]: 'b' == ('a' and 'b')
```

```
Out[6]: True
```

Es *True* debido a que "b" fue el ultimo valor retornado y es cierto que 'b'='b'

---

### 3 Ejercicio 3:

Ordenar tres números enteros en orden ascendente. Escriban un programa que solicite un input de tres números enteros. Convierta el input a tipo int. Use condiciones con "if" para determinar el orden ascendente de los números, y escríbalo a la pantalla (standard output).

```
#PYTHON 2.7.14
```

```
#NUMEROS DE MENOR A MAYOR
```

```
print('----- NUMEROS DE MENOR A MAYOR (DISTINTOS)-----')
```

```
    #SOLICITAR NUMERO Y VERIFICAR QUE SEA ENTERO, SI NO ES ASI, IMPRIMIR MENSAJE
```

```
a=input('Ingresa un numero = ')
```

```
if type(a)!=int:
```

```
    print('El numero debe ser entero!')
```

```
    #COMO a YA TIENE UN TIPO, QUE ES DISTINTO AL ENTERO, SE LE ASIGNA EL VALOR NULO A a
```

```
    #SAQUE LA IDEA DE AQUI: https://stackoverflow.com/questions/19473185/what-is-a-none-value
```

```
    a=None
```

```
    #INSISTIR HASTA QUE SEA ENTERO
```

```
    while type(a)!=int:
```

```
        a=input('Intentalo nuevamente :) , ingresa el numero = ')
```

```
b=input('Ingresa otro numero = ')
```

```
    #SI EL VALOR DE b YA FUE INGRESADO
```

```
if b==a:
```

```
    #INSISTIR PARA QUE SEA DISTINTO
```

```
    while b==a:
```

```

        b=None
        b=input('Ya ingresaste ese valor anteriormente, prueba otro = ')
if type(b)!=int:
    print('El numero debe ser entero!')
    b=None
    while type(b)!=int:
        b=input('Intentalo nuevamente :) , ingresa el numero = ')

c=input('Ingresa el ultimo numero = ')
if c==a or c==b:
    while c==a or c==b:
        c=None
        c=input('Ya ingresaste ese valor anteriormente, prueba otro = ')
if type(c)!=int:
    print('El numero debe ser entero!')
    c=None
    while type(c)!=int:
        c=input('Intentalo nuevamente :) , ingresa el numero = ')
else:
    c=int(c)

    #POSIBLES COMBINACIONES DE LOS NUMEROS INGRESADOS (DISTINTOS) (3!)
x='Los numeros en orden ascendiente son = '
while a<b and a<c:
    if b<c:
        print(x + str(a) + ','+str(b)+','+str(c))
        break
    if c<b:
        print(x + str(a) + ','+str(c)+','+str(b))
        break
    if b==c or c==b:
        print(x + str(a) + ','+str(b)+','+str(c))
        break

while b<a and b<c:
    if a<c:
        print(x + str(b) + ','+str(a)+','+str(c))
        break
    if c<a:
        print(x + str(b) + ','+str(c)+','+str(a))
        break
while c<a and c<b:
    if a<b:
        print(x + str(c) + ','+str(a)+','+str(b))
        break
    if b<a:
        print(x + str(c) + ','+str(b)+','+str(a))
        break

```

---

## 4 Ejercicio 4:

Calcular el factorial de un numero entero. Escriban un programa que solicite input de un número entero. Convierten el numero en tipo int. Use el concepto de iteración (“loops”) para calcular el factorial de este número.

```
#CALCULAR FACTORIAL

#IMPRIMIR EN PANTALLA LO QUE HACE EL PROGRAMA
print('----- FACTORIAL DE UN NUMERO -----')
#MULT ES 1, PARA QUE COMIENCE EL CICLO
mult=1
#SOLICITAR UN NUMERO (ENTERO)
n=input('Ingresa un numero = ')
#MIENTRAS QUE EL NUMERO INGRESADO NO SEA ENTERO, IMPRIMIR EN PANTALLA UN MENSAJE
while type(n)!= int:
    print('La funcion factorial solo admite numeros enteros')
    n=input('Ingresa un numero :) = ')
    #PERO SI EL NUMERO ES ENTERO
if type(n)==int:
    #HACER UN CICLO DONDE SE MULTIPLICAN TODOS LOS NUMEROS ENTEROS ANTERIORES A EL
    #(INCLUYENDOLO)
    for i in range(1,n+1):
        mult=mult*i
print ('El factorial de '+str(n)+' es '+str(mult))
```

---

## 5 Ejercicio 5:

The formula for computing the final amount if one is earning compound interest is given on Wikipedia as

$$A = P \left(1 + \frac{r}{n}\right)^{nt}$$

Where,

- P = principal amount (inicial investment)
- r = annual nominal interest rate (as a decimal)
- n = number of times the interest is compuned per year
- t = number of years

Write a Python program that assigns the principal amount of \$10000 to variable P, assign to n the value 12, and assign to r the interest rate of 8%. Then have the program prompt the user for the number of years t that the money will be compounded for. Calculate and print the final amount after t years

```

#PROGRAMA INTERES COMPUESTO

#ASIGNACION DE VARIABLES

P = 10000
n = 12
r = 0.08

#INGRESAR UN TIEMPO Y REGRESAR EL MONTO ACUMULADO

t=int(input('Ingresa el numero de ano para el cual sera calculada la cantidad de dinero = '))

#CALCULAR EL VALOR ACUMULADO (A)
A= P * (1 + (r/n))**(n * t)

#IMPRIMIR A
print('El valor acumulado en '+str(t)+' anos es = '+str(A))

```

## 6 Ejercicio 6:

Escriba un programa que calcule la raíz de un número  $x$  del tipo float (sea input del usuario) con el siguiente algoritmo:

- Comience con el “guess”  $g = 1$ .
- Si  $\text{abs}((g \oplus g)/x - 1.) < 0.01$ , acabe y diga que  $g$  es el resultado.
- Si no, crea un nuevo supuesto con el promedio de  $g$  y  $x/g$ , usando la asignación  $g = (g + x/g)/2$ .
- Repita esto hasta  $\text{abs}((g \oplus g)/x - 1.) < 0.01$  Use este programa para calcular la raíz de 5489.  
Note: La función abs entrega el valor absoluto de un número:  $\text{abs}(-3.2)=3.2$  y  $\text{abs}(5.4)=5.4$ .

```

#PROGRAMA RAIZ

#COMENZAR CON g = 1

g = 1

#SOLICITAR AL USUARIO UN VALOR FLOTANTE

x = float(input('Ingresa un valor = '))

#MIENTRAS QUE LA SIGUIENTE EXPRESION SEA MAYOR A 0.01
#MIENTRAS MAS PEQUENO SEA EL MIEMBRO DERECHO MAS PRECISO ES EL RESULTADO
#SIN EMBARGO NO SE LA RAZON DE LO ANTERIOR
while abs((g*g)/x-1.)>0.01:
    #g IRA ADQUIRIENDO NUEVOS VALORES
    g=(g+x/g)/2

```

```
#SI LA EXPRESION ANTERIOR ES MENOR AL VALOR ESTABLECIDO ANTES (0.01)
#IMPRIMIR g
if abs((g*g)/x-1.)<0.01:
    print ('La raiz de '+str(x)+' es '+str(g))

#PARA X=5489, el programa arroja 74.2356079049
```

---

## 7 Ejercicio 7:

No fui capaz de anticipar el resultado de todos. Pero el último si ya que se quiere encontrar el resto a una división por 0 y eso en matemáticas es una indeterminación. En Python arroja `ZeroDivisionError`.