----------------------------- ESTRUCTURAS DE CONTROL SELECTIVAS ---------------------------------------------

def obtenerMayor(param1,param2):

if param1 < param2:

print("{} es > {}".format(param2, param1))

obtenerMayor(5, 7) **# 7 es > 5**

obtenerMayor(7, 5) **# No imprime nada**

**# Encadenar más de una condición**

x = y = z = 3

if x == y == z:

print(True) **# True**

----------------------------------------------------------------------------------- if-else ----------------------------------------------------------------------------------------------------

def obtenMayor2(param1,param2):

if param1 < param2:

return param2

else:

return param1

print ("Mayor: {}".format(obtenMayor2 (4, 20) )) **# Mayor: 20**

print ("Mayor: {}".format(obtenMayor2 (11, 6) )) **# Mayor: 11**

**# Simulando operador ternario (Python no tiene éste)**

def obtenMayor3(param1,param2):

valor = param2 if(param1 < param2) else param1

return valor

print ("Mayor: {}".format(obtenMayor3 (11, 6))) **# Mayor: 11**

------------------------------------------------------------------------------ if-elif-else -----------------------------------------------------------------------------------------------

def numeros(num): **# Similar a case o switch**

if num==1:

print ("tu numero es 1")

elif num==2:

print ("el numero es 2")

elif num==3:

print ("el numero es 3")

else:

print ("no hay opción")

numeros(2) **#el numero es 2**

numeros(5) **# no hay opción**

**#se puede evitar la repetición de código del if-elif-else:**

def numeros\_idiom(num): **#Tupla c/opciones válidas**

if num in (1,2,3,4):

print("tu numero es {}".format(num))

else:

print ("{} no es una opcion".format(num))

numeros\_idiom(2) **# tu numero es 2**

numeros\_idiom(5) **# 5 no es una opción**

**# Estructura de control selectiva anidada**

def biggest(a, b, c):

if a > b:

if a > c:

return a

else:

return c

else:

if b > c:

return b

else:

return c

print ("El mayor: {}".format(biggest(7,13,1)))

**# El mayor: 13**

----------------------------------------------------------- ESTRUCTURAS DE CONTROL REPETITIVAS (while / for) --------------------------------------------------------

def cuenta(limite):

i = limite

while true:

print (i)

i = i - 1

if i == 0:

break **# Rompiendo el ciclo**

cuenta(10) **# 10 9 8 7 6 5 4 3 2 1 (vertical)**

def factorial(n):

i = 2

tmp = 1

while i < n+1:

tmp = tmp \* i

i = i + 1

return tmp

print (factorial(4)) **# 24**

**#while-else**

count = 0

while count < 5:

print (count, " < 5")

count = count + 1

else:

print (count, " not < 5")

**# 0 < 5 ... 5 not < 5**

-------------------------------------------------------------------------------- for / range --------------------------------------------------------------------------------

**# range(stop) # range(start, stop[, step])**

for x in range(5): **# range(0,5)**

print(x) **# 0 1 2 3 4 (vertical)**

for x in range(-5, 2):

print(x) **# -5 -4 -3 -2 -1 0 1 2 (vertical)**

for x in range(10, 0, -1):

print(x) **# 10 9 8 7 6 5 4 3 2 1 (vertical)**

for num in ["uno", "dos", "tres", "cuatro"]: **# Lista**

print(num) **# uno dos tres cuatro (vertical)**

for letter in "Python":

print (letter) **# P y t h o n (vertical)**

------------------------------------------------------------------------------ Iteración en diccionarios ------------------------------------------------------------------------

elementos = { 'H': 1, 'HE': 2, 'C': 6 }

for llave, valor in elementos.items():

print(llave, " = ", valor) **# C = 6 H = 1 HE = 2 (vertical)**

for llave in elementos.keys(): **# Sólo llaves**

print(llave) **# C H HE (vertical)**

for valor in elementos.values(): **# Sólo valores**

print(valor) **# 6 1 2 (vertical)**

----------------------------------------------------- función enumerate (si se necesita iterar utilizando un índice) ------------------------------------------------------------

for idx, x in enumerate(elementos): **# Usando índice. Los elementos aparecerán de forma aleatoria (el diccionario es una colección desordenada)**

print("index: {} elemento: {}".format(idx, x)) **# index: 0 elemento: C index: 1 elemento: H index: 2 elemento: HE (vertical)**

---------------------------------------- for-else -----------------------------------------------------------

def cuenta\_idiom(x):

for i in range(x, 0, -1):

print(i)

else:

print("Cuenta finalizada")

cuenta\_idiom(5) **# 5 4 3 2 1 Cuenta finalizada (vertical)**

def cuenta\_idiomv2(limite):

for i in range(limite, 0, -1):

print(i)

if i == 3:

break **# Se rompe el ciclo**

else: **# Corresponde al *for*, no al *if***

print("Cuenta finalizada")

cuenta\_idiomv2(5) **# 5 4 3 (vertical)**

---------------------------------------- utf-8 (para aceptar acentos y más signos de puntuación) / input -----------------------------------------------------------

**# -\*- coding: utf-8 -\*-**

print ("Hola, ¿cómo te llamas?")

nombre = input() **# Entrada en otra línea**

print ("Buen día {}".format(nombre))

**# Entrada en la misma línea**

op = int(input('Opción: ')) **# Toda entrada es cadena, por eso el *cast***