|  |  |
| --- | --- |
|  | **Carátula para entrega de prácticas** |
| Facultad de Ingeniería | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | M.I. Marco Antonio Martinez Quintana |
| *Asignatura:* | estructuras de datos y algoritmos |
| *Grupo:* | 17 |
| *No de Práctica(s):* | 10 |
| *Integrante(s):* | Villanueva Corona Miguel Angel |
| *No. de Equipo de cómputo empleado:* | 36 |
| *No. de Lista o Brigada:* | 41 |
| *Semestre:* | 2020-2 |
| *Fecha de entrega:* | 14 de abril del 2020 |
| *Observaciones:* |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

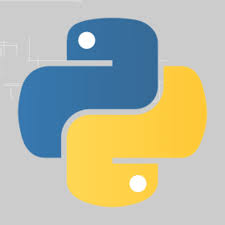
***Objetivos***

Aplicar las bases del lenguaje de programación Python en el ambiente de Jupyter notebook.

***Introducción***

Python es un [lenguaje de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) [interpretado](https://es.wikipedia.org/wiki/Int%C3%A9rprete_(inform%C3%A1tica)) cuya filosofía hace hincapié en la legibilidad de su [código](https://es.wikipedia.org/wiki/Codigo_fuente). ​ Se trata de un lenguaje de programación [multiparadigma](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n#Paradigma_de_programaci%C3%B3n), ya que soporta [orientación a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos), [programación imperativa](https://es.wikipedia.org/w/index.php?title=Programaci%C3%B3n_imperativa&action=edit&redlink=1) y, en menor medida, [programación funcional](https://es.wikipedia.org/wiki/Programaci%C3%B3n_funcional). Es un [lenguaje interpretado](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_interpretado), [dinámico](https://es.wikipedia.org/wiki/Tipado_din%C3%A1mico) y [multiplataforma](https://es.wikipedia.org/wiki/Multiplataforma).

Es administrado por la [Python Software Foundation](https://es.wikipedia.org/wiki/Python_Software_Foundation). Posee una licencia de [código abierto](https://es.wikipedia.org/wiki/C%C3%B3digo_abierto), denominada [Python Software Foundation License](https://es.wikipedia.org/wiki/Python_Software_Foundation_License), que es compatible con la [Licencia pública general de GNU](https://es.wikipedia.org/wiki/GNU_General_Public_License) a partir de la versión 2.1.1, e incompatible en ciertas versiones anteriores.

******

***Desarrollo***

***Estructuras de control selectivas***

***If***

#El funcionamiento de este código es comparar entre dos números uno mayor y otro menor si se invierte el orden no se cuplé la sentencia, si se cumple se imprime un texto en pantalla

def obtenerMayor(param1, param2):

if param1 < param2:

print('{} es mayor que {}'.format(param2, param1))

obtenerMayor(5, 7)

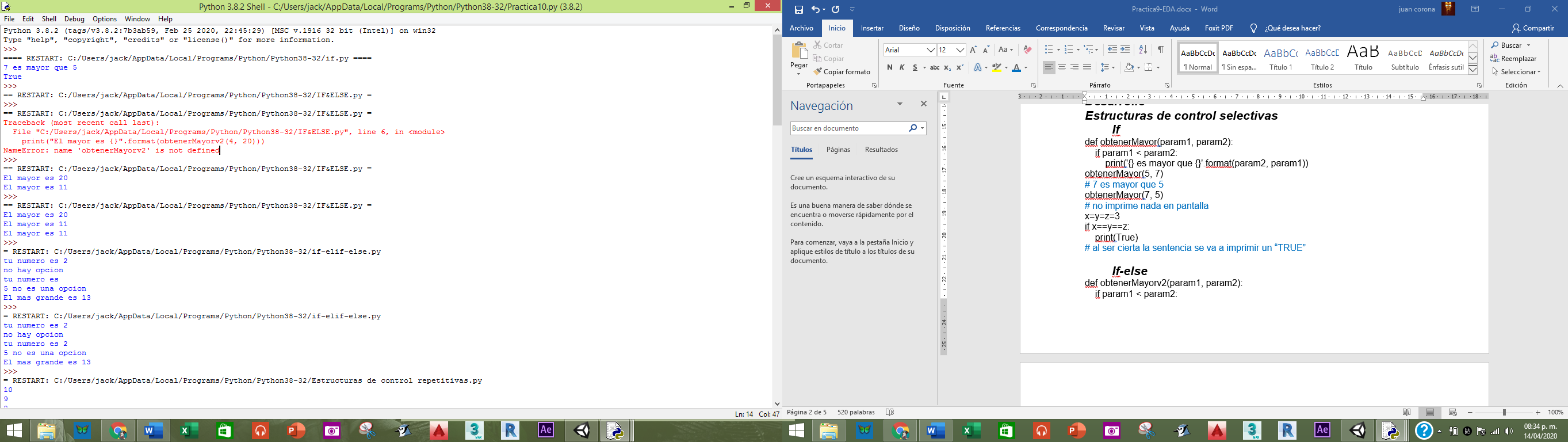
obtenerMayor(7, 5)

#Este código evalúa a las variables para poder cumplir la sentencia si lo cumple no ejecuta las acciones que sean solicitadas

x=y=z=3

if x==y==z:

print(True)



***If-else***

#Los códigos selecciona al número mayor al hacer una comparación entre dos números esto se realiza al evaluar las variables y condiciones si se cumple una de estas se ejecutan las acciones de la que se cumpla.

def obtenerMayorv2(param1, param2):

if param1 < param2:

return param2

else:

return param1

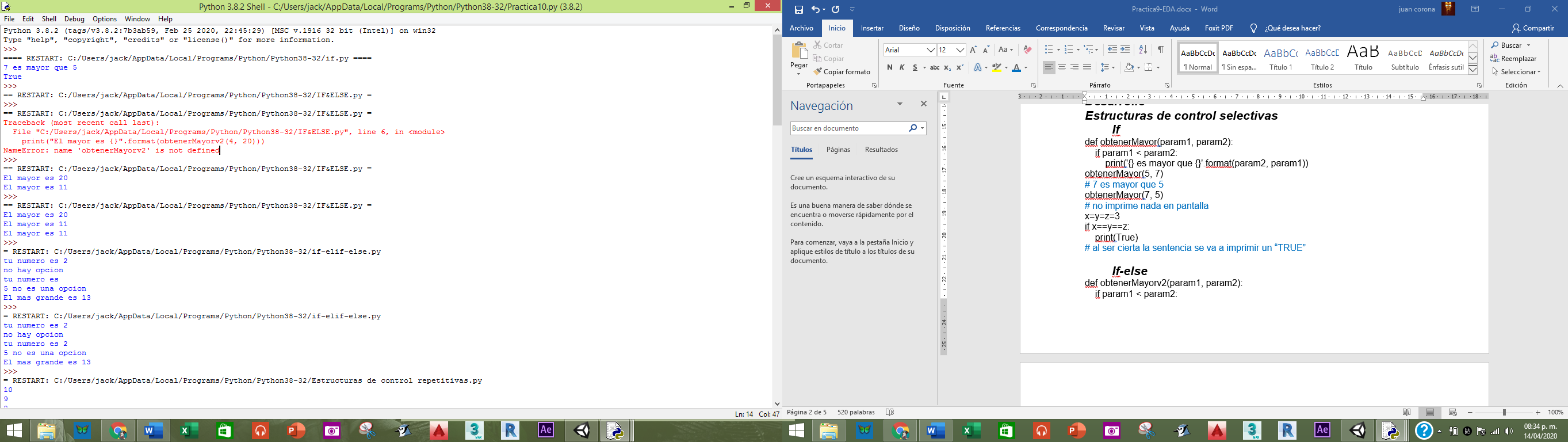
print("El mayor es {}".format(obtenerMayorv2(4, 20)))

print("El mayor es {}".format(obtenerMayorv2(11, 6)))

def obtenerMayor\_idiom(param1, param2):

valor=param2 if (param1 < param2) else param1

return valor

print("El mayor es {}".format(obtenerMayor\_idiom(11, 6)))

***If-elif-else***

#este tipo de sentencia sirve para seleccionar algo de manera múltiple algo así como el swich en el lenguaje de programación en C

def numeros(num):

if num==1:

print("tu numero es 1")

elif num==2:

print("tu numero es 2")

elif num==3:

print("tu numero es 3")

elif num==4:

print("tu numero es 4")

else:

print ("no hay opcion")

numeros(2)

numeros(5)

def numeros\_idiom (num):

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

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

else:

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

numeros\_idiom(2)

numeros\_idiom(5)

def obtenerMasGrande(a,b,c):

if a > b:

if a > c:

return a

else:

return c

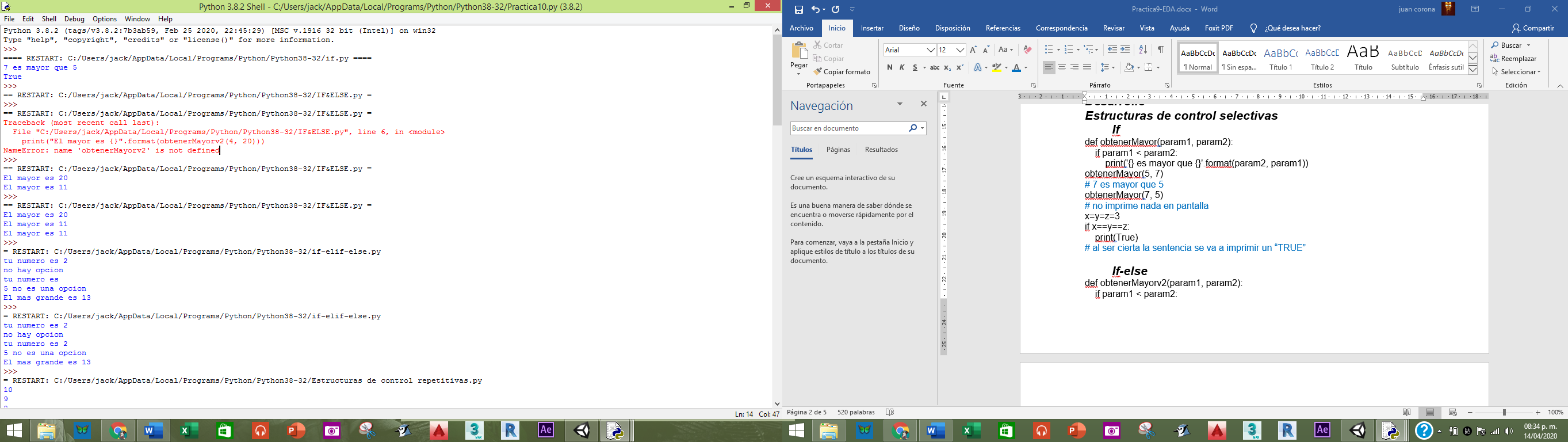
else:

if b > c:

return b

else:

return c

print("El mas grande es {}".format(obtenerMasGrande(7,13,1)))

***Estructuras de control repetitivas***

***Ciclo while***

#Esta sentencia sirve para declarar un ciclo finito o infinito el cual evalúa las variables las cuales sirven para saber si se cumple la acción y hasta que momento se va a cumplir esto facilita al hacer conteos entre otras cosas.

def cuenta(limite):

i=limite

while True:

print(i)

i=i -1

if i==0:

break

cuenta(10)

def factorial(n):

i=2

tmp=1

while i<n+1:

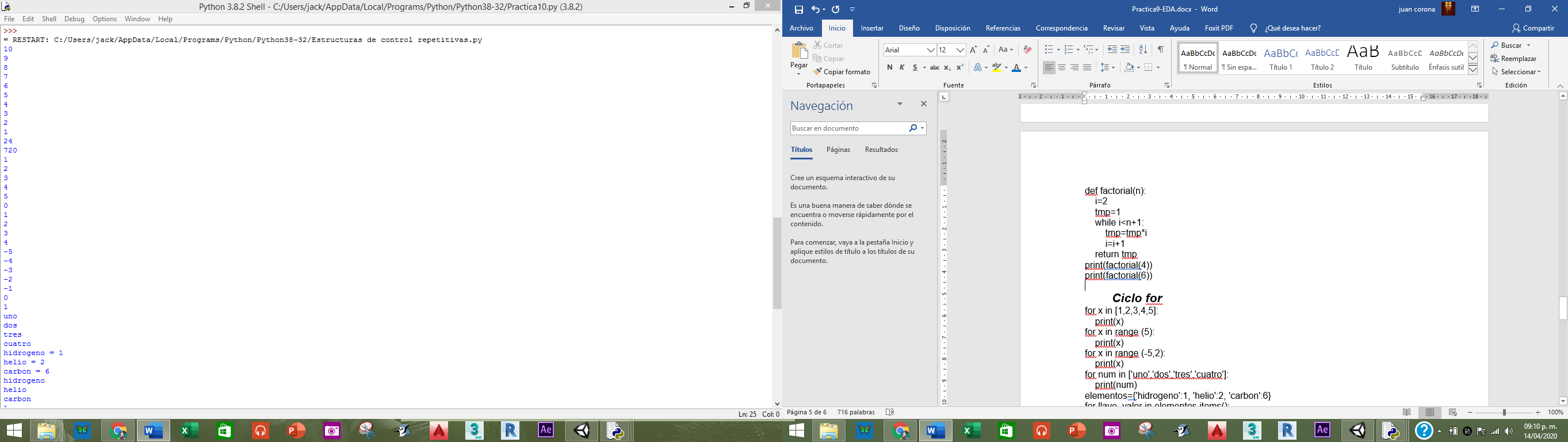
tmp=tmp\*i

i=i+1

return tmp

print(factorial(4))

print(factorial(6))



***Ciclo for***

for x in [1,2,3,4,5]:

print(x)

for x in range (5):

print(x)

for x in range (-5,2):

print(x)

for num in ['uno','dos','tres','cuatro']:

print(num)

elementos={'hidrogeno':1, 'helio':2, 'carbon':6}

for llave, valor in elementos.items():

print(llave, "=", valor)

for llave in elementos.keys():

print(llave)

for valor in elementos.values():

print(valor)

for idx, x in enumerate(elementos):

print('el indice es: {} y el elemento: {}'.format(idx, x))

def cuenta\_idiom(limite):

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

print(i)

else:

print('cuenta finalizada')

cuenta\_idiom(5)

def cuenta\_idiomv2(limite):

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

print(i)

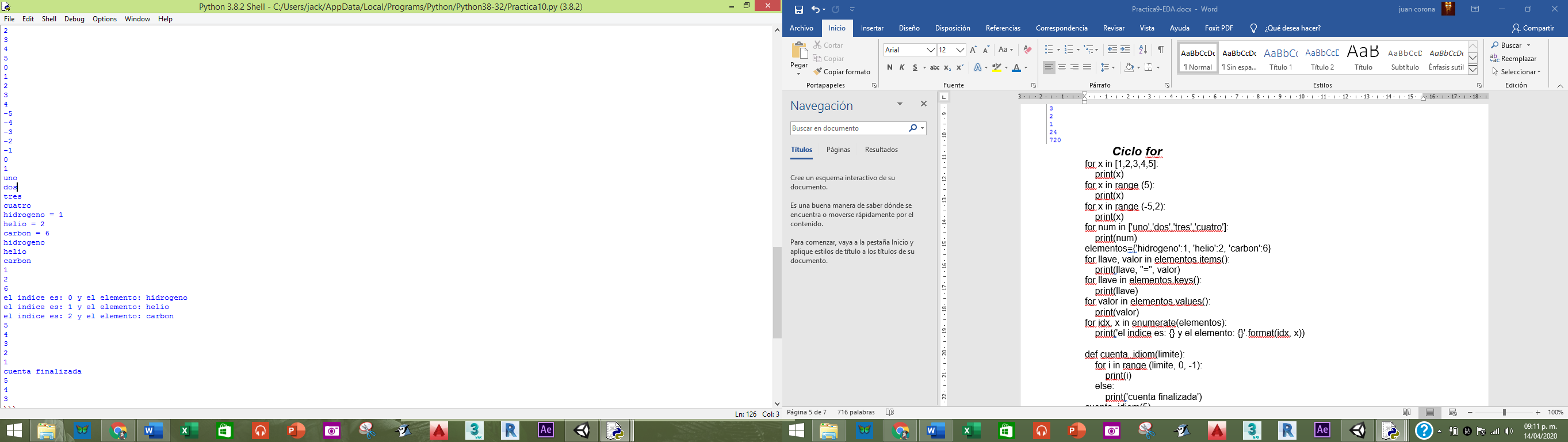
if i==3:

break

else:

print('cuenta finalizada')

cuenta\_idiomv2(5)



***Bibliotecas***

#Las bibliotecas son indispensables para algunas sentencias y sintaxis para que puedan ser ejecutados por el programa

import math

x=math.cos(math.pi)

print(x)

from math import \*

x=cos(pi)

print(x)

from math import cos, pi

x=cos(pi)

print(x)

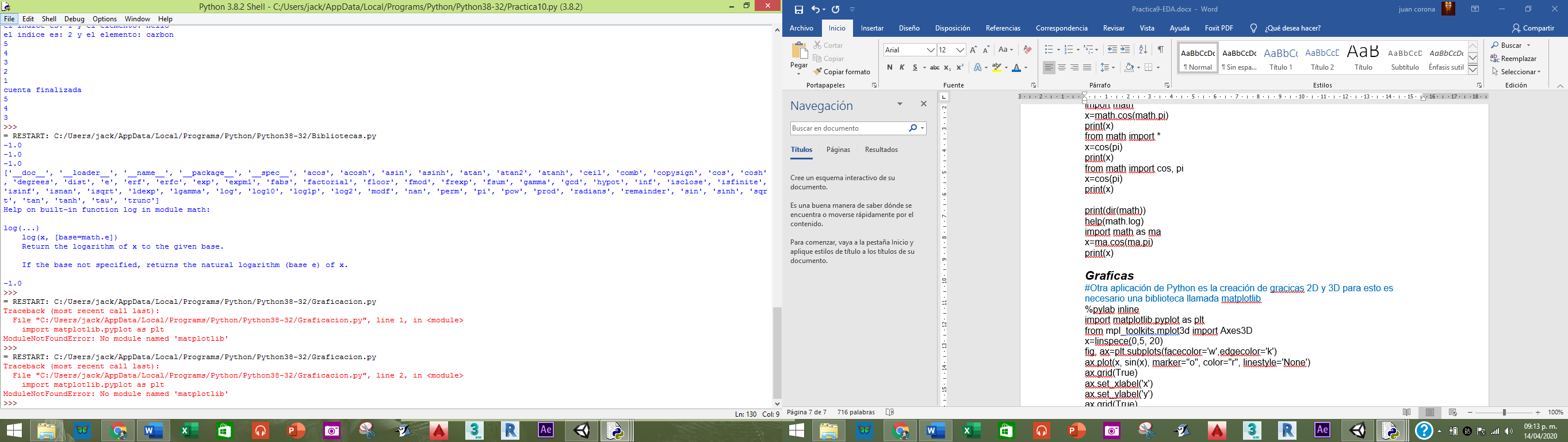
print(dir(math))

help(math.log)

import math as ma

x=ma.cos(ma.pi)

print(x)



***Graficas***

#Otra aplicación de Python es la creación de gracicas 2D y 3D para esto es necesario una biblioteca llamada matplotlib

%pylab inline

import matplotlib.pyplot as plt

from mpl\_toolkits.mplot3d import Axes3D

x=linspece(0,5, 20)

fig, ax=plt.subplots(facecolor='w',edgecolor='k')

ax.plot(x, sin(x), marker="o", color="r", linestyle='None')

ax.grid(True)

ax.set\_xlabel('x')

ax.set\_ylabel('y')

ax.grid(True)

ax.legend(["y=x\*\*2"])

plt.title('puntos')

plt.show()

fig.savefig("grafica.png")

**Conclusión**

Los elementos de selección y los elementos de repetición son de mucha ayuda en la cuestión de la programación ya que suelen ayudar a acortar un programa

***Bibliografía***

<https://es.wikipedia.org/wiki/Python>

<http://lcp02.fi-b.unam.mx/>