**ACTIVITAT AVALUABLE AC4**

**Mòdul:** MP14-Big Data

**UF:** UF1 – Big Data

**Professor:** Albert Guardiola

**Data d’entrega:** 6/11/2022

**Mètode d’entrega:** Clickedu.

Alumno:**ANASS EL MORABIT**

**Resultats de l’aprenentatge:**

RA2. Programa pel Big data a partir de data sources.

**Tareas a realizar:**

**Tarea 1 (1 punto). Comprueba si estos fragmentos de código pueden ejecutarse sin error. En caso de que den error, explica qué tipo de error es, y cuál es su causa.**

1. **dia, temperatura\_maxima = ('lunes', 19, 4)**

**Este codigo es correcto.**

**b) numeros = [1, 2, 3, 4, 5]**

**numeros [10]**

**Este codigo no es correcte, porque la linea “numeros[10] “se utiliza para mostrar el elemento d’indice 10, pero nuesto tabla tiene solo 5 elementos.**

**c) nombre = 'amanda'**

**nombre [0] = 'A'**

**Este codigo es correcto.**

**d) numeros = [1, 2, 3, 4, 5]**

**numeros [3.4]**

**Este codigo estás tratando de acceder a un elemento especifico en la lista “numeros”utilizando la forma de corchetes,pero hay un error en la forma en que estás haciendo,la expreción “numeros[3.4]”no es una forma válida de acceder a un elemento en una lista.**

**e) tupla\_estudiantes = ('Amanda', 'Blue', [98, 75, 87])**

**tupla\_estudiantes[0] = 'Ariana'**

**Este codigo sirve para modificar un elemento de la tuple tupla\_estudiante,lo cual no es possible.las tuplas en python son estructuras de datos inmutables.**

**f) ('Lunes', 87, 65) + 'Martes'**

**Este codigo no es correcto porque no podamos modificar los elementos de la lista a traves de la operacion “+”**

**g) 'A' += ('B', 'C')**

**Este codigo no es correcto porque no podamos modificar los elementos de la lista a traves de la operacion “+=”**

**i) numeros = [1, 2, 3, 4, 5]**

**numeros.index(10)**

**Este codogo no es correcto, porque el comando index se utiliza para accader a un elemento de indice especifico de nuestro lista,pero en este caso nuestro lista tiene solo 5 elementos, por eso no podamos acceder a un elemento de indice 10.**

**j) numeros = [1, 2, 3, 4, 5]**

**numeros.extend(6, 7, 8)**

**Este codigo estas tratando de añadir los elementos 6,7,8 a la lista numeros a traves del comando extend,pero hay un error en la forma en que estás haciendo, porque para añadir muchos elementos a una listaq a traves del extend debemos utilizar los corchetes,no los parenteces.**

**k) numeros = [1, 2, 3, 4, 5]**

**numeros.remove(10)**

**Este codigo es correcto, pero el no sirve para nada, porque el elemento 10 no esta en la lista, por eso no podamos eliminarlo.**

**l) valores = []**

**valores.pop()**

**El comando pop() sirve para eliminar y devolver el último elemento de una lista,pero en este caso, estás intentando usarla en una litsa vacía.**

**Tarea 2 (1 punto). Crea una lista *numeros* que contenga los elementos**

***'1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20'***

**y escribe los siguientes *slices*:**

**a) El tercer número**

**b) Los primeros cinco números.**

**c) La primera mitad de la lista (usa la función len() para saber el tamaño de la lista; y ten en**

**cuenta que los índices de un slice deben ser siempre números enteros).**

**d) Los últimos cinco números.**

**e) Todos los números en orden inverso.**

**f) El tercer número por la cola.**

numeros=[] # crear una lista

for i in range (1,21):# crear una lista

    numeros.append(i)# crear una lista

print(numeros[2])#El tercer número

print(numeros[0:5])#Los primeros cinco números.

print(numeros[0:len(numeros)//2])# la primera mitad de la lista.

print(numeros[15:len(numeros)])#Los últimos cinco números.

print(numeros[len(numeros):0])

print("la lista origin:",numeros)

numeros.reverse()                  #la lista inverse

print("la lista inverse:",numeros) # la lista inverse

print(numeros[2])#el tercer número por la cola.

**Tarea 3 (2 puntos) a)Realiza un fragmento de código que vaya pidiendo al usuario una palabra hasta que contenga la letra secreta. La letra secreta es la “m”, y la sabe el programa pero no el usuario. Utiliza el operador lógico *in*.**

X=input("ingresar una palabra:")

while "m" not in X:

    print("no has introducido la palabra correcta")

    X=input("ingresar una palabra:")

print("has introducido la palabra correcta")

**b) Realiza un fragmento de código que pida al usuario un número de cualquier tamaño,**

**lo convierta en una lista de sus dígitos, y recorra la lista para mostrar por pantalla todos los**

**dígitos por separado, así como cuál es el máximo, el mínimo y la suma de todos ellos.**

**Utiliza la construcción *for...in*, así como las funciones max(), min() y sum().**

Num\_us=int(input("ingresa un numero :"))

numeros=[]

while Num\_us>0:

    X=Num\_us % 10

    numeros.append(X)

    Num\_us=Num\_us // 10

print("la lista:")

print(numeros)

print("el máximo:")

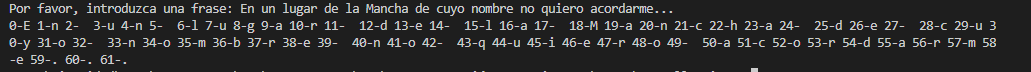
print(max(numeros))

print("el minimo:")

print(min(numeros))

print("la suma:")

print(sum(numeros))

**c)Realiza una función que pida al usuario una frase y que muestre por consola, uno a uno, los caracteres que componen la frase y la posición que ocupa cada uno de ellos. Utiliza la función *enumerate()* en un *for*. Por ejemplo:**

def caracter(X):

    new\_liste=list("X")

    for indice,valor in enumerate(new\_liste):

        print(f"Indice:{indice},Valor:{valor}")

    return

frace=input("introduza una frase:")

caracter(frace)

A screenshot of a computer

Description automatically generated

**Tarea 4 (1 punto). Ejecuta estos dos fragmentos de código. ¿Cuál es la diferencia fundamental que observas?**

***lista2 = sorted(lista1)***

***print(lista1)***

lista1=[3,2,4,65,71,11,0]

lista2=sorted(lista1)

print(lista1)

print(lista2)

A screenshot of a computer

Description automatically generated

***print(lista1.sort())***

***print(lista1)***

***El función sorted se utiliza para ordenar los elementos de una lista,el primero codigo no modifica la lista original”lista1”,en cambio, devuelve una nueva lista ordenada basándose en los elementos de “lista1”,y en el segundo codigo el funcion sort() modifica la lista original “lista1” in-place.No devuelve una nueva lista.***

**Tarea 5 (1 punto). Declara en una sola línea de código una lista que contenga todos las potencias de 2 menores de 536870912. Usa declaración por comprehensión.**

nuevo\_lista=[2\*\*i for i in range(29)]

**Tarea 6 (2 puntos). a)Declara una lista de listas llamada *horario* que contenga los 5x6 horas lectivas semanales.**

**b)Declara una función *rellenarCalendario* que vaya preguntando al usuario qué asignatura se imparte en cada hora. Algunas horas pueden quedar en blanco.**

**c)Utiliza la función *rellenarCalendario* para rellenar la lista *horario* con el horario semanal de 1º de DAW*.***

def rellanCalendario():

    horario = [

    ["Lunes", "", "", "", "", "", ""],

    ["Martes", "", "", "", "", "", ""],

    ["Miércoles", "", "", "", "", "",""],

    ["Jueves", "", "", "", "", "", ""],

    ["Viernes", "", "", "", "", "", ""]

]

    for i,dia in enumerate(horario):

        for j in range(1,len(dia)):

            asignatura=input(f"Ingrese la asignatura para {dia[0]} - Hora {j}:")

            horario[i][j]=asignatura

    return horario

calendria=rellanCalendario()

for i in calendria:

    print(i)

**Tarea 7 (2 puntos). Se trata de realizar un programa que compruebe que en una lista de correos no hay correos inválidos ni repetidos. Para ello, crea una lista de correos electrónicos como la siguiente:**

***lista\_correos = ["albert@gmail.com","rosario@hotmail","rosario@hotmail.com",***

***"javier@gmail.com","rodrigo@etpxavier","albert@gmail.com"]***

**Realiza un programa de Python que haga lo siguiente:**

**PASO I - Muestre por pantalla la lista de correos original, tal cual.**

**PASO II - Elimine de la lista los correos no válidos. Para ello, se debe iterar la lista de correos, y se deben hacer los siguientes pasos para cada correo (tomo como ejemplo el primer correo):**

**2a. Mostrar el correo por pantalla “Correo número 1: albert@gmail.com”**

**2b. Comprobar que en el correo hay una ‘@’ y un ‘.’. Caso negativo: mostrar por pantalla el mensaje: “-Correo no válido: se eliminará de la lista”; y se elimina el correo de la lista.**

**2c. En el caso afirmativo: separar el correo en sus tres partes, y mostrarlas por pantalla: “-albert / gmail / com”.**

**3. Después de completar la iteración de la lista, mostrar por pantalla la lista de nuevo.**

def valido(correo):

    if "@" in correo and "." in correo:

        return True

    else:

         return False

def separar(correo):

    partes=correo.split("@")

    first=partes[0]

    second\_third=partes[1].split(".")

    second=second\_third[0]

    third=second\_third[1]

    return first,second,third

listas\_correos=[

    "anasselmorabit797@gmail.com",

     "albert@gmail.com",

     "juancarlos.com",

     "albert@gmail",

    "albert@gmail.com",

    "albertorodriguez@gmail.com",

     "pepalvaro",

     "anasselmorabit797@gmail.com",

    "albertorodriguez@gmail.com",

     "ANASS ELMORABIT"

]

correos\_validos=[]

for indice,correo in enumerate(listas\_correos):

    print(f"correos numero {indice+1} :{correo}")

    if valido(correo):

        primero,segundo,tersero=separar(correo)

        print(f"{primero} / {segundo} / {tersero}")

        correos\_validos.append(correo)

    else:

        print("Correo no válido: se eliminará de la lista")

print("\nLista de correos válidos:")

**PASO III - Eliminar de la lista los correos repetidos. Sugerencia: crear una lista auxiliar e irla poblando de los correos de la otra lista, siempre que no existan ya en la lista auxiliar. Finalmente, copiar los correos de la lista auxiliar a la lista de correos definitiva.**

**-Mostrar por pantalla la lista de nuevo.**

**Resultado esperado:**

Texto

Descripción generada automáticamente

lista\_copia=[]

for correo in correos\_validos:

    if correo not in lista\_copia:

        lista\_copia.append(correo)

print("Los correos Validos:")

for indice,correo in enumerate(lista\_copia,start=1):

    print(f"Correo número {indice}: {correo}")

A screenshot of a computer program

Description automatically generated