**Instituto Tecnológico de Nuevo Laredo.**



**Ingeniera en Sistemas Computacionales.**

Programación Multiparadigma.

Unidad 1.

**Docente:** Luis Daniel Castillo García.

**Integrantes:**

Humberto Moreno Rivera.

Lizandro Valdez Gutiérrez.

Jose Alberto Varela Hernández.

Septiembre 2022

PRACTICAS

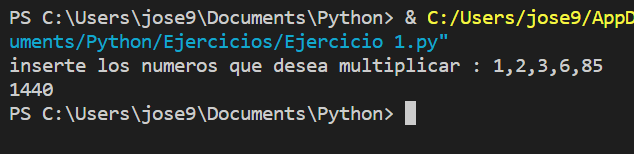
Practica 1: Funciones con n Parámetros.

Escribir un programa que contenga una función que reciba n parámetros de tipo número y calcule el producto total.

Solución:

La solución que le dimos a este problema fue que inicialmente guardamos en una variable los números que se van a ingresar, con un ciclo while recorremos la lista con los números capturados y le quitamos la coma, declaramos una función el cual recibirá la lista de tipo entero y se ira recorriendo e iremos multiplicando los números, al terminar la operación devolvemos el resultado.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | **from** **re** **import** S  **from** **unittest** **import** result  # l=list(input('inserte los numeros que desea multiplicar : '))  # l.extend(input('inserte un numero que quiera multiplicar separados por una coma'))  l=list((input('inserte los numeros que desea multiplicar : ')))  **while** ','**in** l:  l.remove(',')  multiL = list(map(int, l))# La funcion 'map' vuelve una lista de tipo str a int  **def** **multiplicarLista**(multiL):  result=**1**  **for** x **in** multiL:  result = result \* x  **return** result  **print**(multiplicarLista(multiL)) |



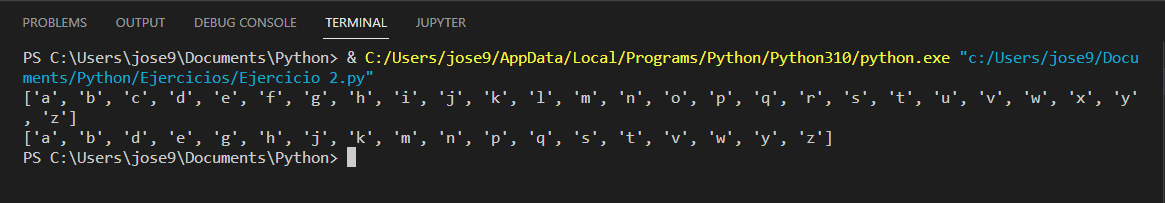
Practica 2: Manejo y manipulación de elementos de una lista.

Escribir un programa que almacene el abecedario en una lista, elimine de la lista las letras que ocupen posiciones de 2, y muestre por pantalla la lista resultante.

Solución:

Para la solución de este problema lo que hicimos fue declaramos una variable y donde almacenaremos la lista del código ascii de nuestro abecedario, con un ciclo for recorremos la lista donde esta nuestro abecedario y validamos que la bandera sea menor que 18, si es asi lo que hacemos en guardamos en una variable nuestra bandera y eliminamos de la lista la letra que contenga el valor de la bandera, le aumentamos 2 y así hasta recorrer toda la lista.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | **from** **copy** **import** deepcopy  **from** **re** **import** A  **import** **string**  Abecedario=list(string.ascii\_lowercase)  nAbecedario=deepcopy(Abecedario)  #IMPRIMIMOS LOS VALORES DEL DICCIONARIO  bandera=**2**  **for** x **in** Abecedario:    **if**(bandera<**18**):  l=bandera  nAbecedario.pop(l)  bandera=bandera+**2**      **else**:  **break**  **print**(Abecedario)  **print**(nAbecedario) |



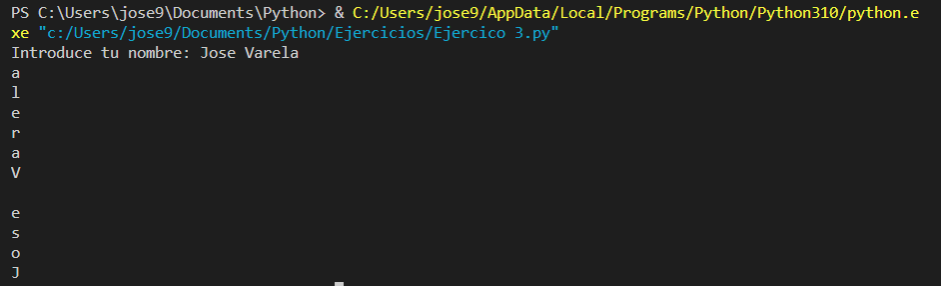
Practica 3: Entrada de datos y manipulación.

Escribir un programa que permite al usuario capturar su nombre completo e imprima su nombre de manera inversa letra por letra.

Solución:

Primero guardamos el nombre que se ingresó, luego declaramos una variable vacía en la cual guardaremos el nombre ya invertido, declaramos un ciclo for con el cual recorremos el nombre letra por letra para irlo concatenando, primero concatenamos la letra y despues la variable en donde estamos almacenando el nombre invertido, una vez terminado eso declaramos otro for pero este solo será para imprimir el nombre letra por letra.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | nombre = input("Introduce tu nombre: ")  nombre\_invertido = ""  **for** letra **in** nombre:  nombre\_invertido = letra + nombre\_invertido  **for** letra **in** nombre\_invertido:  **print**(letra) |



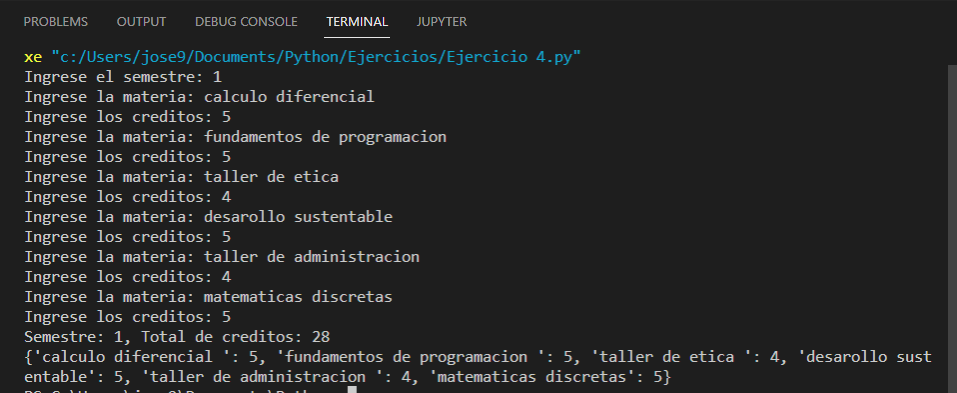
Practica 4: Entrada de datos y estructuración.

Revisar su retícula para escribir un programa que cree un diccionario vacío para que el usuario capture las materias y créditos de su semestre preferido (inferior a 8vo) al final imprimir en el formato “{asignatura}” tiene “{crédito}” crédito. Y la suma de todos los créditos del semestre.

Solución:

Lo primero que hicimos fue declarar una variable en donde almacenaremos el semestre que ingreso el usuario, declaramos nuestro diccionario y una variable en donde guardaremos la suma de los créditos totales. Validamos que el semestre ingresado no sea mayo a 8vo, agregamos otra validación en donde si el semestre es mayor a 2do guardamos un cierto numero de materias en una variable ya que en la retícula el primer semestre solo son 6 materias y si es mayor a 2 semestre las materias son 7.

Una vez hecho esas validaciones con un ciclo while vamos a iterar sobre el numero de materias que almacenamos anteriormente para que ingrese el nombre de la materia y los créditos de dicha materia, por cada iteración iremos sumando los créditos de las materias y almacenando en el diccionario el nombre y los créditos de la materia, y vamos incrementando de uno en uno hasta que el ciclo llegue al total de materias.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | **import** **string**  semestre = int(input("Ingrese el semestre: "))  diccionario = {}  creditos\_totales = **0**  i =**1**  **if** semestre > **8**:  **print**("No se puede ingresar un semestre mayor a Octavo")  **if** semestre > **2** :  num\_materias = **6**  **else** :  num\_materias = **7**  **while** i < num\_materias:  materia = input("Ingrese la materia: ")  creditos = int(input("Ingrese los creditos: "))  creditos\_totales += creditos  diccionario[materia] = creditos  i += **1**  **print**("Semestre: " + str(semestre)+", Total de creditos: "+ str(creditos\_totales))  **print**(diccionario) |

Practica 5: Manejo de información.

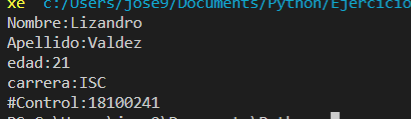
Escribir una función que reciba n parámetros de llave valor e imprima la información en formato “{llave}”:” {valor}”

Solución:

Definimos una función que recibe n parámetros despues dentro de esa función declaramos un for el cual nos ayudara a recorrer los parámetros que haya recibido e imprimiremos su llave y su valor.

Declaramos una variable para guardar los parámetros que le mandaremos a la función.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | **def** **imprimirFormato**(\*\*id):  **for** llave, valor **in** id.items():  **print**(f'{llave}:{valor}')  datos = {'Nombre':'Lizandro','Apellido':'Valdez', 'edad':**21**,'carrera':'ISC',  '#Control':'18100241'}  imprimirFormato(\*\*datos) |



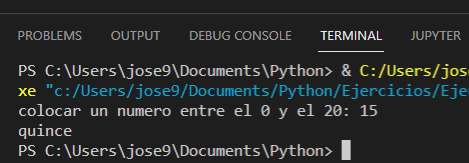
Practica 6: Razonamiento y prueba de código.

Escribir un programa que reciba un numero entre el 0 y el 20 e imprimir el numero en letra, no utilizar condicionales, máximo 5 líneas de código.

Solución:

La solución a este problema es muy sencilla, declaramos una lista en donde almacenamos los números escritos y ya solo cuando el usuario ingrese el número que quiere imprimir le mandamos el índice a la lista del número que queremos que muestre.

|  |  |
| --- | --- |
| 1  2  3  4  5 | numero = ['zero','uno','dos','tres','cuatro','cinco','seis','siete','ocho','nueve','diez'  ,'once','doce','trece','catorce','quince','dieciseis',  'diecisiete','dieciocho','diecinueve','veinte']  a = int (input("colocar un numero entre el 0 y el 20: ") )  **print**(numero[a]) |



Practica 7: Formateo y conversiones.

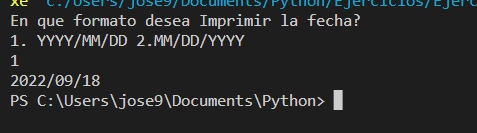
Escribir un programa que muestre un menú con 2 opciones la primera opción “1.-Imprimir YYYT/MM/DD” la segunda “2.-Imprimir MM/DD/YYYY” una vez seleccionada la opción imprimir la fecha de hoy en el formato seleccionado.

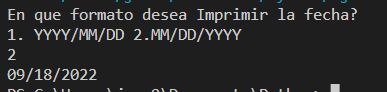
Solución:

En esta ocasión para hacer el programa ocupamos hacer una importación de la librería datetime para poder utilizar sus propiedades y métodos de fecha, una vez importada la librería declaramos una variable en donde almacenaremos la opción que seleccione el usuario.

Hacemos una validación en donde si el usuario selecciono la opción número uno se imprimirá la fecha en el formato correspondiente en caso contrario la fecha se imprimirá en el segundo formato.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | **from** **datetime** **import** datetime  f=input("Que formato de hora quieres imprimir 1.-(YYYY/MM,DD), 2.-(MM/DD/YYYY)")  fecha= datetime.now()  **if** f==**1**:  format=fecha.strftime("%Y/%m/%d")  **print**(format)  **else**:  format=fecha.strftime("%m/%d/%Y")  **print**(format) |





Practica 8: Resumen y multi-solucion.

**8.1.-Definir una clase usuario que contenga como atributos:**   
Usuario   
Contraseña   
Rol   
Nombre   
CURP   
Ciudad

**8.2.-Realizar un programa que contenga el siguiente menú**

1.- Registro   
2.- Inicio de sesión   
3.- Salida

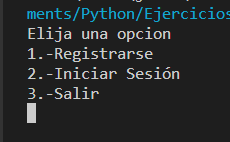
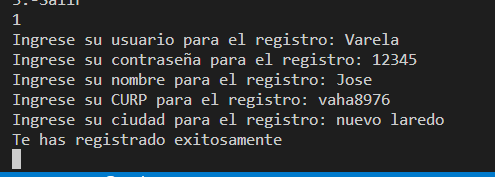
La opción de registro solicitara al usuario registrarse solicitando la información de los atributos la clase   
exceptuando el atributo Rol que por defecto será rol cliente, no se permitirán usuarios con CURP   
repetido en caso de mostrar mensaje de “El usuario ya existe”   
La opción de inicio de sesión permitirá al usuario introducir sus credenciales al ser correctas desplegar   
en pantalla la información del usuario de lo contrario mostrar mensaje de “datos incorrectos“   
**8.3.- Declarar un usuario con rol “Administrador” el cual al momento de iniciar sesión despliegue la   
información de todos los usuarios registrados al momento.**

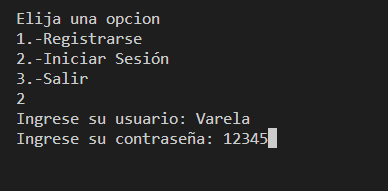
Solución:

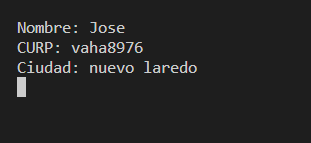
Para este problema lo que hicimos fue crear una clase con los atributos correspondientes, despues creamos 2 métodos el cual uno es para registrar un usuario y el otro es para iniciar sesión.

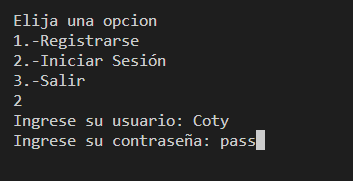
Para el método de registro declaramos las variables necesarias para almacenar los datos que ingresara el usuario y además se agregó una validación para que no existan usuarios repetidos validando que el curp no sea igual. En el segundo método que es el que se usa para iniciar sesión se declararon variables para almacenar el usuario y el password, y validamos que tipo de usuario es ya que si el usuario tiene el rol de administrador se le desplegara una lista con todos los usuarios registrados en cambio si el usuario es de tipo cliente solo podrá ver en pantalla su propio usuario.

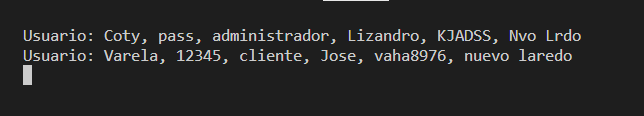
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76 | **import** **os**  #8.1=========================  **class** **Usuario**:  **def** **\_\_init\_\_**(self, usuario, password,rol, nombre, curp, ciudad) -> None:  self.usuario = usuario  self.password = password  self.rol = rol  self.nombre = nombre  self.curp = curp  self.ciudad = ciudad  **def** **\_\_str\_\_**(self) -> str:  **return** f'Usuario: {self.usuario}, {self.password}, {self.rol}, {self.nombre}, {self.curp}, {self.ciudad}'  #8.1=========================  #8.2=========================  usuarios = []  miUsuario = Usuario("Coty","pass","administrador","Lizandro", "KJADSS", "Nvo Lrdo") #8.3  usuarios.append(miUsuario)  **def** **Registro**():  usuario\_r = input("Ingrese su usuario para el registro: ")  password = input("Ingrese su contraseña para el registro: ")  nombre = input("Ingrese su nombre para el registro: ")  curp = input("Ingrese su CURP para el registro: ")  ciudad = input("Ingrese su ciudad para el registro: ")  agrega = '1'  **for** usuario **in** usuarios:  **if** curp == usuario.curp:  input("Este curp ya esta registrado")  agrega = '0'  **if** agrega == '1':  miUsuario = Usuario(usuario\_r,password,"cliente",nombre, curp, ciudad)  usuarios.append(miUsuario)  input("Te has registrado exitosamente**\n**")  os.system ("cls")  **def** **IniciarSesion**():  usuario\_ini\_ses = input("Ingrese su usuario: ")  password\_inis\_ses = input("Ingrese su contraseña: ")  os.system ("cls")  encontrado = "1"  **for** usuario **in** usuarios:  #8.3  **if** usuario\_ini\_ses == usuario.usuario **and** password\_inis\_ses == usuario.password **and** usuario.rol =="administrador":  **for** usuario **in** usuarios:  **print**(usuario)  encontrado = "1"  **break**  **else**:  **if** usuario\_ini\_ses == usuario.usuario **and** password\_inis\_ses == usuario.password:  **print**(f'Nombre: {usuario.nombre}**\n**CURP: {usuario.curp}**\n**Ciudad: {usuario.ciudad}')  encontrado = "1"  **break**  **else**:  encontrado = "0"  **if** encontrado == "0":  **print** ("Datos incorrectos")  input()  os.system ("cls")  respuesta = "0"  **while** respuesta != "3" :  respuesta = input("Elija una opcion **\n**1.-Registrarse**\n**2.-Iniciar Sesión **\n**3.-Salir**\n**")  **if** respuesta **in** ("1","2","3"):  match respuesta:  case "1":  Registro()    case "2":  IniciarSesion()  **else**:  **print**("Ingrese una de las opciones correctas")  input()  os.system ("cls")  os.system ("cls") |







Usuario administrador:



Conclusiones:

Enlace de github:

https://github.com/HMR17100861/Equipo-Moreno-Varela-Valdez