

Ejercicios

- 1) Dadas 10 letras ingresadas, determinar cuál es vocal y cuál es consonante.
- 2) Calcular y Mostrar la serie Fibonacci hasta el 100. (cada número es la suma de los dos precedentes, comenzando por 0 y 1).
- 3) Escriba un programa que pida una cantidad de segundos y que escriba cuántos minutos y segundos son.
- 4) Escriba un programa que pida una distancia en pies y pulgadas y que escriba esa distancia en centímetros. Se recuerda que un pie son doce pulgadas y una pulgada son 2,54 cm.
- 5) Escriba un programa que pida una temperatura en grados Celsius y que escriba esa temperatura en grados Fahrenheit. Se recuerda que la relación entre grados Celsius (C) y grados Fahrenheit (F) es la siguiente: $F = 1,8 * C + 32$
- 6) Escribir un programa que pregunte al usuario por el número de horas trabajadas y el costo por hora. Después debe mostrar por pantalla el pago que le corresponde.
- 7) Calcular la Distancia recorrida por un auto a velocidad constante y en un tiempo determinado. Tener en cuenta la fórmula de Movimiento Rectilíneo uniforme (Distancia es igual a Velocidad por Tiempo).
- 8) Calcular el promedio de nota obtenido por un alumno, teniendo 7 notas en el año.
- 9) Escribir un programa que pregunte al usuario una cantidad a invertir, el interés anual y el número de años, y muestre por pantalla el capital obtenido en la inversión.
- 10) Se pide obtener la calificación final de un examen, teniendo un nro de respuesta correctas que suman 4 puntos c/u, un nro de respuestas incorrectas que restan 1 punto por c/u y un nro de respuestas en blanco.
- 11) Una juguetería tiene mucho éxito en dos de sus productos: payasos y muñecas. Suele hacer venta por correo y la empresa de logística les cobra por peso de cada paquete así que deben calcular el peso de los payasos y muñecas que saldrán en cada paquete a demanda. Cada payaso pesa 11,2 kg y cada muñeca 7,5 kg. Escribir un programa que lea el número de payasos y muñecas vendidos en el último pedido y calcule el peso total del paquete que será enviado.
- 12) Se pide obtener que tipo de triángulo es. Sabiendo que ingresan 3 datos correspondientes al largo de cada lado. Recordar: el triángulo equilátero tiene los 3 lados iguales, el isósceles 2 lados iguales y el escaleno los 3 lados distintos.

Ejercicios

- 13) Definir una función que calcule la longitud de una lista o una cadena dada.
- 14) Definir una función inversa() que calcule la inversión de una cadena. Por ejemplo la cadena "estoy probando" debería devolver la cadena "odnaborp yotse".
- 15) Definir una función es_palindromo() que reconoce palíndromos (es decir, palabras que tienen el mismo aspecto escritas invertidas), ejemplo: es_palindromo ("radar") tendría que devolver True.
- 16) Dado un texto ingresado convertir a mayúscula la primer letra de cada palabra.
- 17) Escriba una función es_bisiesto() que determine si un año determinado es un año bisiesto. Un año bisiesto es divisible por 4, pero no por 100. También es divisible por 400.
- 18) Escribe un programa que pida dos palabras y diga si riman o no. Si coinciden las tres últimas letras tiene que decir que riman. Si coinciden sólo las dos últimas tiene que decir que riman un poco y si no, que no riman.
- 19) Escribe una función llamada "elimina_duplicados" que elimine los elementos duplicados en una lista y los devuelva en una nueva lista.
- 20) De la galería de productos, el usuario introducirá el código y el número de unidades del producto que desea comprar. El programa determinará el total a pagar, como una factura.
Una variante a este ejercicio que lo haría un poco más complejo sería dar la posibilidad de seguir ingresando diferentes códigos de productos con sus respectivas cantidades, y cuando el usuario desee terminar el cálculo de la factura completa con todas sus compras.
- 21) Este programa muestra primero el listado de categorías de películas y pide al usuario que introduzca el código de la categoría de la película y posterior a ello pide que el usuario introduzca el número de días de atraso, y así se muestra al final el total a pagar.
- 22) Escribe una función que lea las palabras de un archivo de texto (texto.txt) y construya una lista donde cada palabra es un elemento de la lista.

Ejercicios

23) Escribir un programa para gestionar un listado telefónico con los nombres y los teléfonos de los clientes de una empresa. El programa incorpora funciones para crear el fichero con el listado si no existe, para consultar el teléfono de un cliente, añadir el teléfono de un nuevo cliente y eliminar el teléfono de un cliente. El listado debe estar guardado en el fichero de texto listado.txt donde el nombre del cliente y su teléfono deben aparecer separados por comas y cada cliente en una línea distinta. Variante guardar con formato de diccionario.

24) Diseña un programa que cuente el número de caracteres de un fichero de texto, incluyendo los saltos de línea. (El nombre del fichero se pide al usuario por teclado.)

25) Haz un programa que, dada una palabra y un nombre de fichero, diga si la palabra aparece o no en el fichero. (El nombre del fichero y la palabra se pedirán al usuario por teclado.)

26) Haz un programa que, dado un nombre de fichero, muestre cada una de sus líneas precedida por su número de línea. (El nombre del fichero se pedirá al usuario por teclado.)

27) Haz una función que, dadas la ruta de un fichero y una palabra, devuelva una lista con las líneas que contienen a dicha palabra.

Diseña a continuación un programa que lea el nombre de un fichero y tantas palabras como el usuario desee (utiliza un bucle que pregunte al usuario si desea seguir introduciendo palabras). Para cada palabra, el programa mostrará las líneas que contienen dicha palabra en el fichero.

28) Haz un programa que muestre por pantalla la línea más larga de un fichero. Si hay más de una línea con la longitud de la más larga, el programa mostrará únicamente la primera de ellas. (El nombre del fichero se pedirá al usuario por teclado.)

```
# determinar si una letra es vocal o consonante.
```

```
#!/usr/bin/env python
```

```
# -*- coding: utf-8 -*-
```

```
# con validacion en el ingreso comprobamos si la letra es vocal o consonante
```

```
listaLetras = ["a","b","c","d","e","f","g","h","i","j","k","l","m","n","ñ","o","p","q","r","s","t","u","v","w","x","y","z","A","B","C","D","E","F","G","H","I","J","K","L","M","N","Ñ","O","P","Q","R","S","T","U","V","W","X","Y","Z"]
```

```
listaVocal = ["a","e","i","o","u","A","E","I","O","U"]
```

```
ok = "no"
```

```
letra = str()
```

```
cant = 0
```

```
while cant < 11:
```

```
    while ok == "no":
```

```
        letra = str(input("Ingrese una letra: "))
```

```
        if letra in listaLetras:
```

```
            print("es una letra del abecedario")
```

```
            if letra in listaVocal:
```

```
                print(letra, " es una Vocal")
```

```
                print("Programa finalizado")
```

```
                ok = "si"
```

```
            else:
```

```
                print(letra, " es una Consonante")
```

```
                print("Programa finalizado")
```

```
                ok = "si"
```

```
        else:
```

```
            print("Debes escribir una letra.")
```

```
            cant += 1
```

```
            ok= "si"
```

```
            print(cant,"intentos")
```

```
            if cant == 11:
```

```
                ok = "si"
```

```
                print("Se agotaron las posibilidades")
```

```
            else:
```

```
                ok = "no"
```

```
                print("Intente de nuevo")
```

2)

serie Fibonacci

serie = [0,1]

uno = 0

dos = 1

tres = 0

while tres < 100:

tres = uno + dos

if tres < 100:

serie.append(tres)

uno = dos

dos = tres

print(serie)

3)

Escriba un programa que pida una cantidad de segundos y que escriba cuántos minutos y segundos son.

```
ingSegundos = int(input("Ingrese cantidad de segundos: "))
```

```
restSegundos = 0
```

```
minutos = ingSegundos // 60    # calcula el cociente usando //
```

```
restSegundos = ingSegundos % 60 # calculo el resto usando %
```

""" otra manera:

```
valSegundos = minutos * 60
```

```
if valSegundos != ingSegundos:
```

```
    restSegundos = ingSegundos - valSegundos"""
```

```
print(ingSegundos, "corresponde a", minutos, "minutos y", restSegundos,  
      "segundos")
```

4)

#Escriba un programa que pida una distancia en pies o pulgadas y que # escriba esa distancia en centímetros.

Se recuerda que un pie son doce pulgadas y una pulgada son 2,54 cm

```
ingOpcion = int(input("Ingrese 1 para Pies, 2 para Pulgadas: "))
```

```
ingPies = input("Ingrese distancia:")
```

```
resCent = float(ingPies)
```

```
if ingOpcion == 1:
```

```
    resCent = float(ingPies) * 30.48
```

```
else:
```

```
    resCent = float(ingPies) * 2.54
```

```
print(ingPies, " corresponde a ", resCent, " centímetros.")
```

5)

```
# Escriba un programa que pida una temperatura en grados Celsius y que escriba
# esa temperatura en grados Fahrenheit. Se recuerda que la relación entre grados
# Celsius (C) y grados Fahrenheit (F) es la siguiente:  $F = 1,8 * C + 32$ 
#
valido = True
while valido == True:
    try: # se puede usar try/catch errores
        gCelsius = int(input("Ingrese temperatura en Grados Celsius: "))
        valido = False
    except ValueError:
        print("Debes escribir un número.")
gFar = 1.8 * gCelsius + 32
print("Equivale en Fahrenheit a: ",gFar)
```


6)

Escribir un programa que pregunte al usuario por el número de horas trabajadas y
el costo por hora. Después debe mostrar por pantalla el pago que le corresponde.

#

valido = True

while valido == True:

try:

numHoras = int(input("Ingrese cantidad de horas trabajadas por día: "))

valido = False

except ValueError:

print("Debes escribir un número.")

while valido == False:

try:

pagoJornal = int(input("Ingrese el valor de la hora: "))

valido = True

except ValueError:

print("Debes ingresar un valor positivo.")

calcJornal = float(pagoJornal * numHoras)

print("El valor del jornal es: ", calcJornal)

7)

Calcular la Distancia recorrida por un auto a velocidad constante y en un tiempo

determinado. Tener en cuenta la fórmula de Movimiento Rectilíneo uniforme:

(Distancia es igual a Velocidad por Tiempo).

#

valido = True

while valido == True:

try:

ingVelo = int(input("Ingrese la velocidad: "))

valido = False

except ValueError:

print("Debes escribir un número.")

while valido == False:

try:

ingTiempo = int(input("Ingrese el tiempo expresado en Horas: "))

valido = True

except ValueError:

print("Debes escribir un número.")

calDistancia = ingVelo * ingTiempo

print("Recorrerá KM ", calDistancia, " a ", ingVelo, " en ", ingTiempo, "horas.")

8)

Calcular el promedio de nota obtenido por un alumno, teniendo 7 notas en el año.

nota = 0

promedio = 0

cant = 0

while cant < 7:

 while True:

 try:

 nota = float(input("Ingrese nota a promediar:"))

 promedio = promedio + nota

 cant = cant + 1

 break

 except ValueError:

 print("Debe ingresar un valor numerico")

promedio = round(promedio / 7,2)

print("El promedio es: ",promedio)

9)

Escribir un programa que pregunte al usuario una cantidad a invertir, el interés
anual y el número de años, y muestre por pantalla el capital obtenido en la
inversión.

```
cantImp = float(input("Ingrese importe a invertir: "))
cantInt = int(input("Ingrese Interes Anual: "))
cantAnios = float(input("Ingrese cantidad de Años: "))
cant = 0
while cant < cantAnios:
    interes = cantInt * cantImp / 100
    print(interres)
    cantImp = cantImp + interes
    cant += 1
print("Capital resultante: ",cantImp)
```

10)

Se pide obtener la calificación final de un examen, teniendo un

nro de respuesta correctas que suman 4 puntos c/u,

un nro de respuestas incorrectas que restan 1 punto por c/u y

un nro de respuestas en blanco.

```
respOk = int(input("Ingrese cantidad de respuestas correctas:"))
```

```
respMal = int(input("Ingrese cantidad de respuestas Incorrectas:"))
```

```
calif = respOk * 4 - respMal
```

```
print("La calificación final es:", calif)
```

11)

```
#!/usr/bin/env python
```

```
# -*- coding: utf-8 -*-
```

```
# calcular el peso del paquete que debemos enviar de acuerdo a la cantidad de muñecas y payasos
```

```
pay = 11.2
```

```
mun = 7.5
```

```
while True:
```

```
    try:
```

```
        payasos = int(input("Ingrese cantidad de payasos vendidos: "))
```

```
        munecas = int(input("Ingrese cantidad de muñecas vendidas: "))
```

```
    except ValueError:
```

```
        print("El peso no puede ser menor a 0")
```

```
    if payasos < 0:
```

```
        print("El peso de payasos no puede ser menor a 0: ")
```

```
    elif munecas < 0:
```

```
        print("El peso de munecas no puede ser menor a 0")
```

```
    else:
```

```
        break
```

```
pedido = (payasos * pay) + (munecas * mun)
```

```
print("El peso del paquete es: ", pedido)
```

12) Tipo de triángulo según 3 valores ingresados.

```
#!/usr/bin/env python
```

```
# -*- coding: utf-8 -*-
```

```
while True:
```

```
    try:
```

```
        ladoA = int(input("ingrece la medida del lado 1 en cm: "))
```

```
        ladoB = int(input("ingrece la medida del lado 2 en cm: "))
```

```
        ladoC = int(input("ingrece la medida del lado 3 en cm: "))
```

```
    except ValueError:
```

```
        print("La medida debe ser un numero mayor a 0")
```

```
    else:
```

```
        if ladoA <= 0:
```

```
            print("La medida 1 no puede ser menor a 0.")
```

```
        elif ladoB <= 0:
```

```
            print("La medida 2 no puede ser menor a 0.")
```

```
        elif ladoC <= 0:
```

```
            print("La medida 3 no puede ser menor a 0.")
```

```
        else:
```

```
            break
```

```
if ladoA == ladoB and ladoA == ladoC:
```

```
    print("El triagulo es un equilatero")
```

```
elif ladoA == ladoB or ladoA == ladoC or ladoB == ladoC:
```

```
    print("El triangulo es un isósceles")
```

```
else:
```

```
    print("el triangulo es un escaleno")
```

13)

Definir una función que calcule la longitud de una lista o una cadena dada

auxTexto = "Quiero saber cuantos caracteres estoy usando"

mesLista = ["Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio", "Agosto", "Sept", "Oct", "Nov", "Dic"]

cont = 0

cont2 = 0

for cant in auxTexto:

 cont += 1

print("La cadena tiene:",cont,"caracteres.")

cont = 0

for cant in mesLista:

 varAux = mesLista[cant]

 cont2 = 0

 for cuento in varAux:

 cont2 += 1

 print(mesLista[cant],"tiene:", cont2,"caracteres.")

 cont = cont + 1

print("La lista tiene: ", cont, "items.")

14)

```
#!/usr/bin/env python
```

```
# -*- coding: utf-8 -*-
```

```
# invertimos un string
```

```
cadena = input("Ingrese un texto para invertir su orden: ")
```

```
def largo_cadena(cadena):
```

```
    cont = 0
```

```
    for i in cadena:
```

```
        cont += 1
```

```
    return cont
```

```
invertida = ""
```

```
cont = largo_cadena(cadena)
```

```
print(cont)
```

```
indice = -1      # para tomar el último elemento de la cadena
```

```
while cont >= 1:
```

```
    invertida = invertida + cadena[indice]
```

```
    indice = indice + (-1)
```

```
    cont -= 1
```

```
print(invertida)
```

15)

```
#!/usr/bin/env python
```

```
# -*- coding: utf-8 -*-
```

```
# Invertimos un string para ver si al derecho y alreves es igual, ejemplo: radar, ala, somos, seres, ojo, reconocer, orejero, arenera
```

```
# alreves = cadena[::-1] # esto invierte el string, no vale usarlo aqui
```

```
cadena = str(input("Ingrese una palabra: "))
```

```
invertida = ""
```

```
cont = len(cadena)
```

```
indice = -1 # para tomar el último elemento de la cadena
```

```
while cont >= 1:
```

```
    invertida = invertida + cadena[indice]
```

```
    indice = indice + (-1)
```

```
    cont -= 1
```

```
if cadena == invertida:
```

```
    print(cadena,"Es palindromo")
```

```
else:
```

```
    print(cadena,"NO es palindromo")
```

16)

```
#!/usr/bin/env python
```

```
# -*- coding: utf-8 -*-
```

```
# convertimos la primer letra de cada palabra en mayuscula, no usar El método capitalize() devuelve un string en el que
```

```
# solo la primera letra estará escrita en mayúsculas, tampoco usar title() que pone en mayusc la primer letra de cada palabra
```

```
def titulo(cadena):
```

```
    nueva = ""
```

```
    inicioPalabra = True                #indica el inicio de una palabra
```

```
    for caracter in cadena:
```

```
        if not caracter.isalpha():
```

```
            nueva=nueva+caracter
```

```
            inicioPalabra = True
```

```
        else:
```

```
            if inicioPalabra:
```

```
                nueva = nueva + caracter.upper()
```

```
                inicioPalabra = False        #ya no es el inicio de una palabra
```

```
            else:
```

```
                nueva = nueva + caracter.lower()
```

```
    return nueva
```

```
cadena = input("Ingreso una oracion:")
```

```
print(titulo(cadena))
```

17)

```
#!/usr/bin/env python
```

```
# -*- coding: utf-8 -*-
```

```
def es_bisiesto(anio):
```

```
    if anio % 4 == 0 and (not(anio % 100 == 0)):
```

```
        print("El año", anio, "es un año bisiesto ")
```

```
    elif anio % 400 == 0:
```

```
        print("El año", anio, "es un año bisiesto ")
```

```
    else:
```

```
        print("El año", anio, "no es bisiesto")
```

```
#
```

```
print("Comprueba años bisiestos")
```

```
anio = int(input("Escriba un año y le dire si es bisiesto: "))
```

```
es_bisiesto(anio)
```

18)

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# ingresan dos palabras para saber si riman, primero
Validaciones
print("Vamos a ver si estas palabras riman.")
while True:
    try:
        uno = input("Dime la primera palabra: ")
    except ValueError:
        print("Ingresa un valor Alfabético.")
    else:
        if uno.isalpha():
            break
while True:
    try:
        dos = input("Dime la segunda palabra: ")
    except ValueError:
        print("Ingresa un valor Alfabético.")
    else:
        if dos.isalpha():
            break
```

```
# Ahora hago las comprobaciones
if len(uno) < 3 or len(dos) < 3:
    print("Las palabras tienen menos de 3 letras")
elif uno[-3:] == dos[-3:]:
    print(uno, "y", dos, "Riman")
elif uno[-2:] == dos[-2:]:
    print(uno, "y", dos, "Riman un poco")
else:
    print(uno, "y", dos, "No riman")
```

19)

```
#!/usr/bin/env python
```

```
# -*- coding: utf-8 -*-
```

```
# Escribe una función llamada "elimina_duplicados" que elimine los elementos duplicados en una lista
```

```
# y los devuelva en una nueva lista.
```

```
lista = ["Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio", "Marzo", "Agosto", "Sept", "Junio", "Oct", "Nov", "Dic",  
"Dic"]
```

```
listaElim = []
```

```
repeAux = ""
```

```
print("Esta es la lista original: ", lista)
```

```
resultado = []
```

```
duplicados = []
```

```
def elimina_duplicados(lista):
```

```
    for nombre in lista:
```

```
        if nombre not in resultado:
```

```
            resultado.append(nombre)
```

```
        else:
```

```
            duplicados.append(nombre)
```

```
elimina_duplicados(lista)
```

```
print("Lista modificada: ", resultado)
```

```
print("Nombre removidos: ", duplicados)
```

20)

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
```

```
def main():
```

```
----print "ELIJA EL PRODUCTO DESEADO"
----print ""
----print "PRODUCTO CODIGO"
----print ""
----print "CAMISA..... 1"
----print "CINTURON..... 2"
----print "ZAPATOS..... 3"
----print "PANTALON..... 4"
----print "CALCETINES..... 5"
----print "FALDAS..... 6"
----print "GORRAS..... 7"
----print "SUETER..... 8"
----print "CORBATA..... 9"
----print "CHAQUETA..... 10"
----print ""
```

```
----relacion = {1:35, 2:10, 3:50, 4:40, 5:5, 6:20, 7:7, 8:15, 9:10, 10:35}
```

```
----codigo = input("INTRODUZCA EL CODIGO: ")
----print ""
----print "EL PRECIO ES: $", relacion[codigo]
----cantidad = input("INTRODUZCA EL NUMERO DE UNIDADES: ")
----print ""
```

```
----precio_t = float(relacion[codigo] * cantidad)
```

```
----print "EL TOTAL A PAGAR ES: $", precio_t
```

```
main()
```

21)

```
# -*- coding: utf-8 -*-
#!/usr/bin/env Python
# vamos a calcular el costo del alquiler de una pelicula
diccio = {
'Categoria' : ["Acccion", "Aventuras", "Drama", "Comedia", "Historica"],
'Recargo' : [10, 10, 5, 15, 2],
}
cate = 0
listaCateg = diccio["Categoria"]
print("***** Categorías *****")
for indi in listaCateg: # muestro el menú de categorías
    print(cate, "-", indi)
    cate += 1
while True: # valido el ingreso de categoría
    try:
        tipo = int(input("Elija la categoría: "))
        if tipo < 0 or tipo > 4:
            print("La categoría debe ser de 0 a 4.")
        else:
            break
    except ValueError:
        print("Debe ingresar un valor numérico.")
atraso = int(input("Ingrese los días de atraso en la entrega: "))
costo = 200 + ((diccio["Recargo"][tipo] * 200 / 100) * atraso)
print("El valor del alquiler es $ 200, \nmas los días de atraso, Deberá abonar: $", costo)
```


Aprendemos sobre diccionarios

```
MiAgenda = {
'Nombre' : "Maria",
'Apellido' : "Perez",
'Alias' : "Mery",
'Padres' : ["Maria Gomes", "Carlos Perez"], #<-- esto es una lista
'Edad' : 55,
'Genero' : "Femenino",
'Estado Civil' : "Soltera",
'Hijos' : ("Juan", "Pedro", "Tomas", "Carla"), # esto es una tupla
'Mascotas' : "Gatos",
'Nombres de mascotas' : ["Mishu", "Pompon"]} #<-- esto es una
lista
}
#print(MiAgenda)
#print(MiAgenda["Apellido"])
#print(MiAgenda["Padres"][0:2])
MiAgenda["Ingresos"] = 20000 # agregamos la clave Ingresos
con un valor
#print(MiAgenda["Ingresos"])
# creamos la clave Salario y eliminamos Ingresos:
MiAgenda["Salario"] = MiAgenda.pop("Ingresos")
#print(MiAgenda["Salario"])
#print(MiAgenda.get("Apellido", "No existe Nombre"))
#claves = MiAgenda.keys()
#print(claves)
#valores = MiAgenda.values()
#print(valores)
```

```
lista = [("semana","lunes"),("Meses","Junio"),("Campo", "nuevo")]
#nuevoDiccio = {
#"semana": "lunes",
#"Meses": "Junio"}
# estoy creando un dic tomando el primer valor como clave y el segundo como
valor:
nuevoDiccio = {a:b for a, b in lista}
#print(nuevoDiccio)
#claves = nuevoDiccio.keys()
#print(claves)
#valores = nuevoDiccio.values()
#print(valores)
# creamos un diccionario con listas funcionadas usando zip()
nLista = ["Pedro", "Juan", "Tomas", "Pablo"]
eLista = [32,40,28,38]
diccio = dict(zip(nLista, eLista))
newDicc = {
'Meses': ["Enero", "Febrero", "Marzo"],
'Edades': dict(zip(nLista, eLista)),
}
print(newDicc)
```

Aprendemos manejo de Archivos

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# ejemplo manejo de archivos
# ArchiObj = open("ArchiUno.txt", "w") #Creamos el archivo
# ArchiObj.write("Creamos un archivo de texto en python de forma clásica
# escribiendolo")
# ArchiObj.close() #Cerramos el archivo
#

ArchiObj = open("ArchiUno.txt", "r")
ArchiObj.seek(10)
veo = ArchiObj.read()
ArchiObj.close()
print("Uno: ", ArchiObj)

# creamos archivo nuevo abriendolo de escritura y le grabamos contenido
with open("NuevoArchi.txt", "w") as ObjArchi:
    ObjArchi.write("Le estoy escribiendo algo de contenido")
    ObjArchi.write("segunda linea, escribiendo algo.")
    ObjArchi.write("tercera linea, será suficiente?.")
    ObjArchi.close()

# abrimo archivo de lectura para leerlo completo
with open("NuevoArchi.txt", "r") as Archi:
    contenido = Archi.read()
    Archi.close()
print("NuevoArchi: ", contenido)
```

```
# abrimos archivo de lectura para leerlo renglon por renglon
with open ("CostumbresArgentinas.txt", "r") as archi: # Abrimos en modo Read (r)
    lineaA = archi.readline() # Lo abrimos utilizando el metodo read
    lineaB = archi.readline() # segunda linea
    archi.close()
# lineaC = archi.readlines() # lee todo
print("Cancion: ", lineaA) # mostramos primer linea
print("Interprete: ", lineaB) # mostramos segunda linea
# print("Todo: ", lineaC)

def recuArchi(bArchi):
    estribillo = list(enumerate(bArchi))
    for ind, linea in estribillo:
        if ind >= 1 and ind <= 4:
            print(linea)

with open ("CostumbresArgentinas.txt", "r") as bArchi: #Abrimos como modo
read (r)
    estribillo=(linea for ind,linea in enumerate(bArchi) if ind>=6 and ind<=9)
    for linea in estribillo:
        print (linea)
    for ind,linea in enumerate(bArchi): print(ind,linea)
    bArchi.close()
listaLenguajes = ["Cobol", "Java", "C", "Clipper", "Natural"]
# print(list(enumerate(listaLenguajes)))
for ind, lenguaje in enumerate(listaLenguajes):
    print(ind, lenguaje)
```

Manejo de Archivos:

usamos **re-write**

```
with open ("CostumbresArgentinas.txt", "r+") as cArchi: #Abrimos a modo Reescribir (r+)
```

```
    contenido = cArchi.read()      # Lo abrimos utilizando read y guardamos contenido
    final_del_archivo = cArchi.tell() #(OPCIONAL)Definimos variable con el puntero al final
```

```
    cArchi.write("Que buena cancion!!") # Escribimos la nueva línea
    cArchi.seek(final_del_archivo)    #(OPCIONAL)
    contenido_editado = cArchi.read() #(OPCIONAL) Guardamos nueva línea con read()
    print (contenido_editado)        #(OPCIONAL) mostramos contenido nuevo
    cArchi.close()
```

usamos **append**

```
with open ("CostumbresArgentinas.txt", "a") as dArchi: #Abrimos el usando el modo (a)
    dArchi.write("Usamos append para agregar una linea mas.") # Escribimos la nueva línea
```

```
    dArchi.close()
```

```
with open("CostumbresArgentinas.txt", "r") as fArchi:
```

```
    nuevo = fArchi.read()
    fArchi.close()
    print("uso Append: ", nuevo)
```

usamos **writelines** para agregar desde una lista

```
Lista = ["Cancion \n", "muy pero muy \n", "argento"]
```

```
with open ("CostumbresArgentinas.txt", "r+") as gArchi: #Abrimos usando Reescribir (r+)
```

```
    contenido = gArchi.read()      # Lo abrimos utilizando read
    gArchi.writelines(Lista)       # Escribimos la Lista en el archivo linea a linea
    gArchi.close()
```

```
with open("CostumbresArgentinas.txt", "r") as hArchi:
```

```
    newContenido = hArchi.read()
    print("Agrego Desde lista: ", newContenido)
```

ejemplo de apertura sin archivo, capturamos el error y #pedimos ubicación del mismo.

Abrimos utilizando Reescribir (r+)

```
dirarchivo = "ana.txt"
```

```
while True:
```

```
    try:
```

```
        with open (dirarchivo, "r+") as fichero:
```

```
            contenido = fichero.read()                print(contenido)
            break
```

```
        except:
```

```
            print("Error al intentar Abrir")
```

```
            print("No se encuentra el archivo", dirarchivo, ", especifique su ubicación.")
```

```
            dirarchivo = (input("Nombre del Archivo: "))
```

#Para leer línea por línea iterativamente:

```
with open("CostumbresArgentinas.txt", "r") as archi:
```

```
    for linea in archi:
```

```
        print(linea)
```

Una vez que estés leyendo un archivo línea a línea, podés hacer otras operaciones, como separar los datos dentro de una línea con el método `split()`: `renglon = linea.split(',')` □ si los campos están separados por coma.

Recuperatorio Parcial:

```
#!/usr/bin/env python
```

```
# -*- coding: utf-8 -*-
```

```
'''
```

Se ingresan 10 palabras y se guardan en una lista, se pide:

- ordenarlas alfabéticamente

- mostrar la lista ordenada.

```
listalImpares = [13,7,1,17,5,9,11,19,3,15]
```

Sobre la lista de números impares buscar el nro mayor y ordenarla de menor a mayor.

Se pide mostrar la lista ordenada y el nro mayor encontrado.

Aclaración: se puede usar el método de ordenamiento en la lista de palabras, no así, en la lista de números.

```
'''
```

```
#----- Armado y ordenamiento de Lista de Palabras -----
```

```
cant = 1
```

```
miLista = []
```

```
while cant <= 10:
```

```
    while True:
```

```
        try:
```

```
            palabra = input("Ingrese una palabra: ")
```

```
            if palabra.isalpha():
```

```
                miLista.append(palabra)
```

```
                break
```

```
            else:
```

```
                print("Debe ingresar texto alfabético.")
```

```
        except ValueError:
```

```
            print("Se esperaba una palabra.")
```

```
        cant = cant + 1
```

```
print("Se ha ingresado: ", miLista)
```

```
ordLista = sorted(miLista)
```

```
print("Te ordeno la lista:", ordLista)
```

```
#----- Buscar mayor y ordenar de menor a mayor la
```

```
lista -----
```

```
listalImpares = [13,7,1,17,5,9,11,19,3,15]
```

```
canti = len(listalImpares) - 1
```

```
cuento = 0
```

```
nuelmpares = []
```

```
mayor = 0
```

```
for nro in listalImpares:
```

```
    if nro > mayor:
```

```
        mayor = nro
```

```
print("El Numero Impar mayor es: ", mayor)
```

```
auxLista = listalImpares
```

```
while cuento <= canti:
```

```
    menor = 0
```

```
    for nro in auxLista:
```

```
        if menor == 0:
```

```
            menor = nro
```

```
        else:
```

```
            if nro < menor:
```

```
                menor = nro
```

```
nuelmpares.append(menor)
```

```
auxLista.remove(menor)
```

```
cuento = cuento + 1
```

```
print("Lista de Impares ordenada: ", nuelmpares)
```