

Introducción a la programación con Python Archivos, Módulos

Alexis Rodríguez

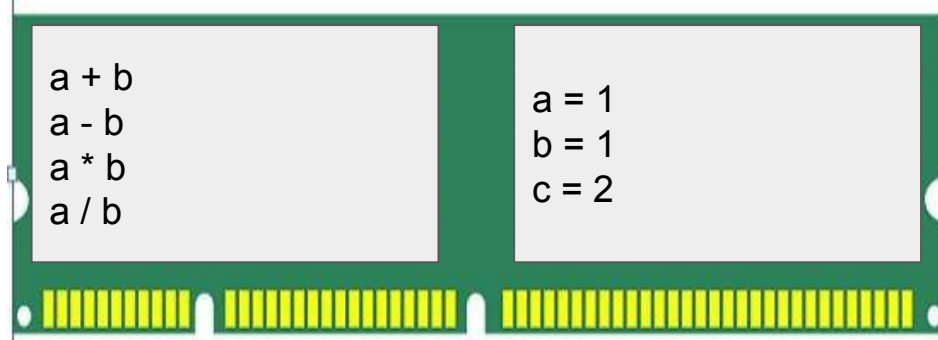
Marcel Morán C

Esquema

- ¿Qué es un archivo?
- Ruta de archivos
- Sintaxis para Escribir/Leer a un archivo
- Modulos

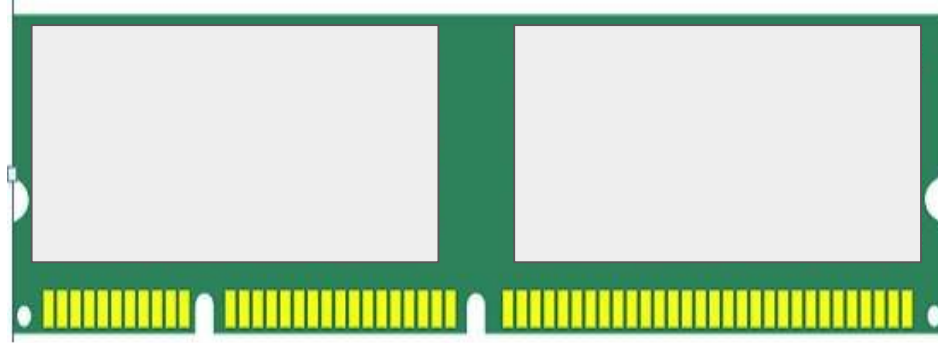
Guardando datos

- Variables se asignan a tipos de datos
- Datos en RAM (tipo interno de memoria)
- La velocidad de la RAM permite impacta en el tiempo de ejecución de instrucciones



Guardando datos

- Variables se asignan a tipos de datos
- Datos en RAM (tipo interno de memoria)
- La velocidad de la RAM permite impacta en el tiempo de ejecución de instrucciones
- Datos no se conservan al terminar ejecución



Guardando datos

- Variables se asignan a tipos de datos
- Datos en RAM (tipo interno de memoria)
- La velocidad de la RAM permite impacta en el tiempo de ejecución de instrucciones
- Una vez la finalizada la ejecución del program, tipos datos se eliminan del sistema
- Disco duros, pen drives, disquetes (tipo externo de memoria)
- Lentos pero seguro (la mayoría del tiempo)



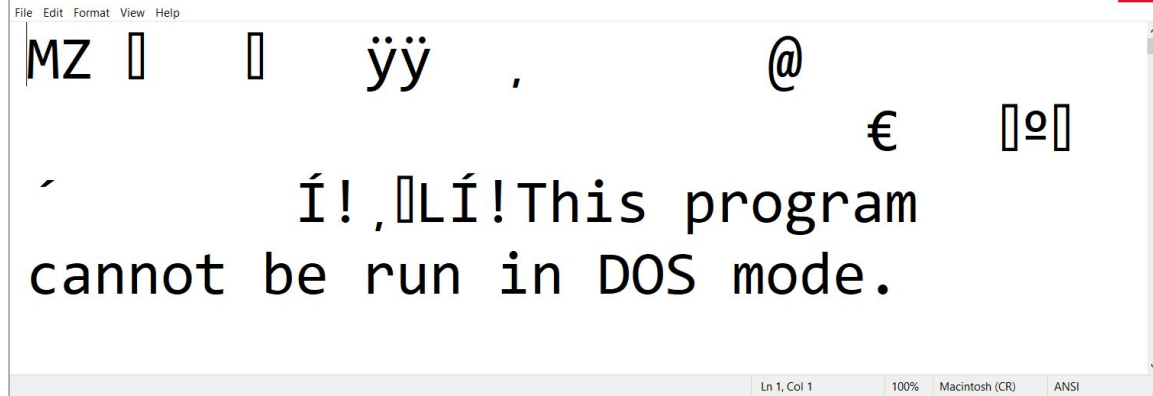
Guardando datos

- Variables se asignan a tipos de datos
- Datos en RAM (tipo interno de memoria)
- La velocidad de la RAM permite impacta en el tiempo de ejecución de instrucciones
- Una vez la finalizada la ejecución del program, tipos datos se eliminan del sistema
- Disco duros, pen drives, disquetes (tipo externo de memoria)
- Lentos pero seguro (la mayoría del tiempo)



¿Qué es un archivo?

- Información guardada en memoria externa de sistema
- Guardado de información texto o binario
- código binario no es comprensible para nosotros
- Archivos tienen dos propiedades (el nombre del archivo y su ruta)



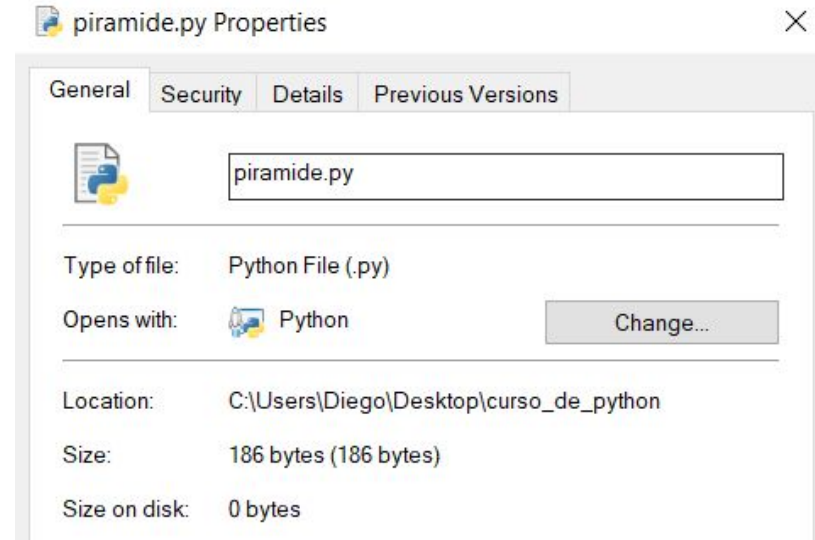
The image shows a screenshot of a text editor window with a menu bar (File, Edit, Format, View, Help). The text content is as follows:

```
MZ  [ ]  yy ,      @  
                                     €  [o]  
      Í! , [LÍ! This program  
cannot be run in DOS mode.
```

The status bar at the bottom indicates "Ln 1, Col 1", "100%", "Macintosh (CR)", and "ANSI".

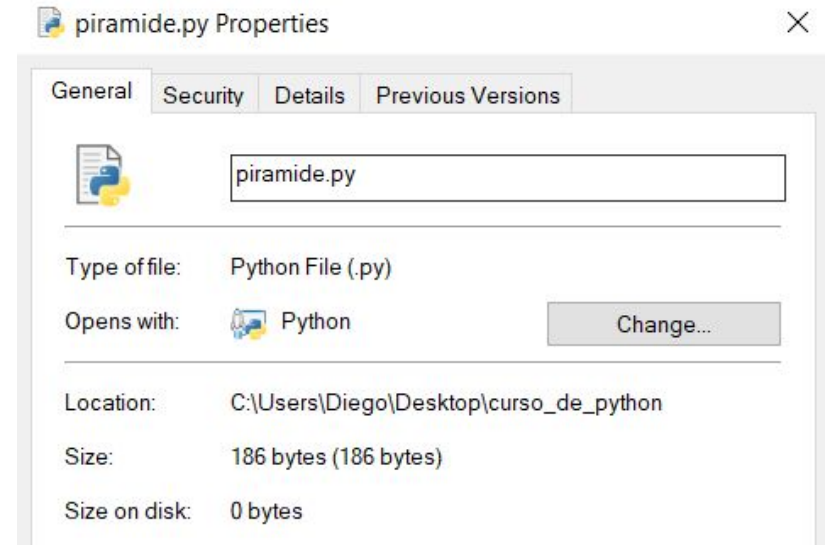
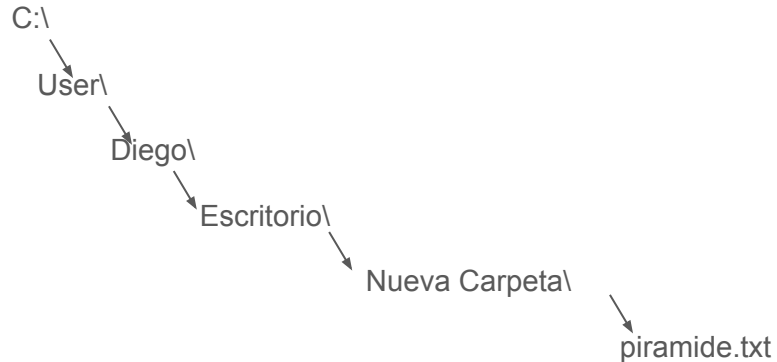
Ruta de archivos

- Permite referenciar el acceso a los archivos de dispositivos de información
- Organizado por el sistema de ficheros



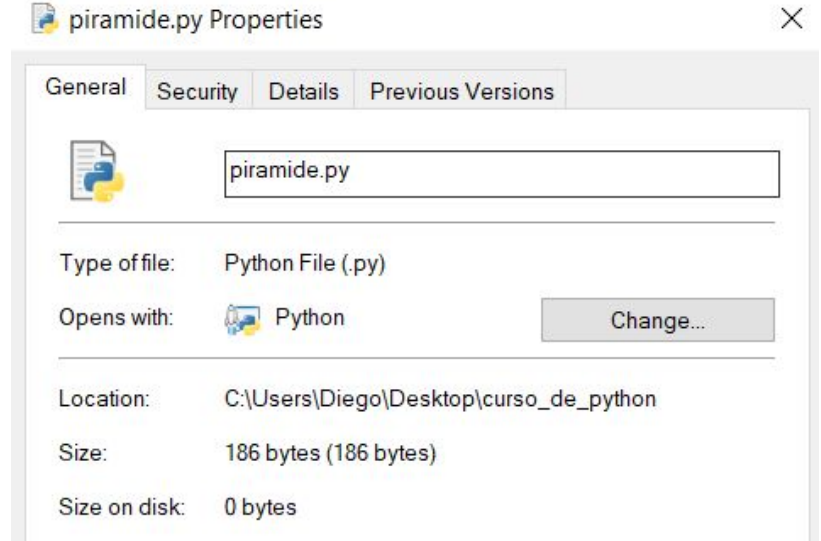
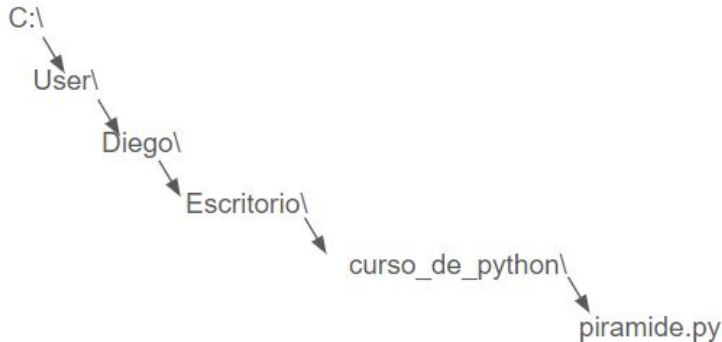
Ruta de archivos

- Permite referenciar el acceso a los archivos de dispositivos de información
- Organizado por el sistema de ficheros



Ruta de archivos

- Permite referenciar el acceso a los archivos de dispositivos de información
- Organizado por el sistema de ficheros
- Jerarquía de sistema, comenzando desde C:\

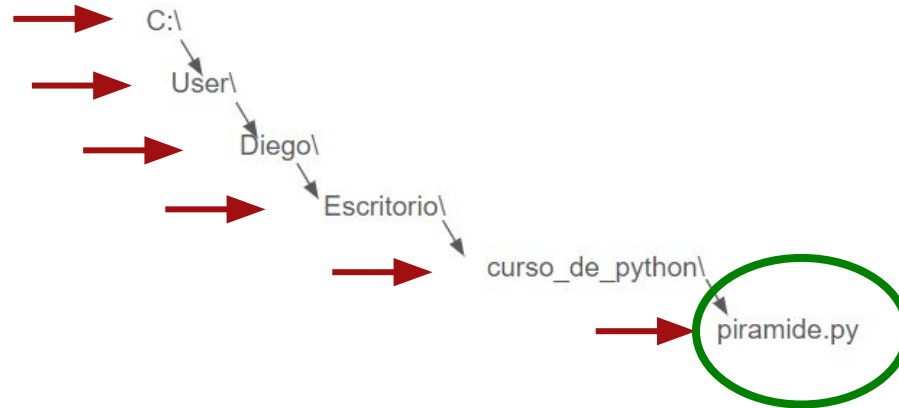


Ruta de archivos

- Permite referenciar el acceso a los archivos de dispositivos de información
- Organizado por el sistema de ficheros
- Jerarquía de sistema, comenzando desde raíz C:\
- Rutas Absolutas comienzan desde la raíz del sistema C:\Users\usuario\Escritorio\programa.py
- Rutas relativas comienzan desde el directorio actual ..\ , .\, .\Escritorio\program.py

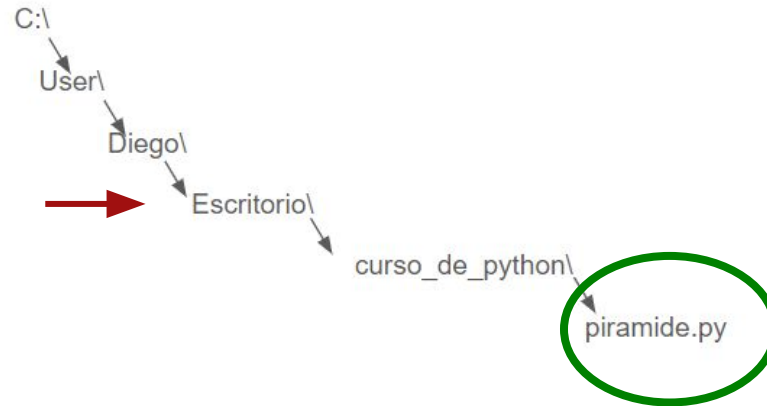
Ruta de archivos

- Permite referenciar el acceso a los archivos de dispositivos de información
- Organizado por el sistema de ficheros
- Jerarquía de sistema, comenzando desde raíz C:\
- Rutas Absolutas comienzan desde la raíz del sistema
C:\Users\usuario\Escritorio\programa.py
- Rutas relativas comienzan desde el directorio actual `..\` , `..\` , `..\Escritorio\programa.py`



Ruta de archivos

- Permite referenciar el acceso a los archivos de dispositivos de información
- Organizado por el sistema de ficheros
- Jerarquía de sistema, comenzando desde raíz C:\
- Rutas Absolutas comienzan desde la raíz del sistema C:\Users\usuario\Escritorio\programa.py
- Rutas relativas comienzan desde el directorio actual ..\ , .\, **.\Escritorio\program.py**

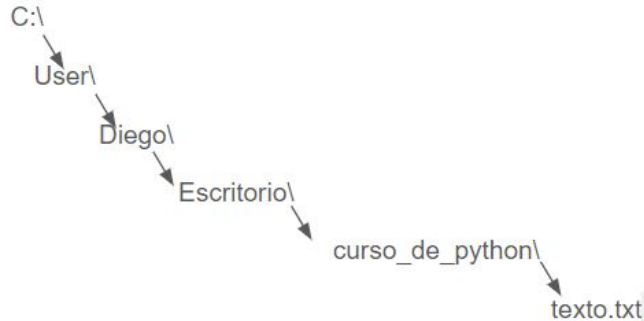


Sintaxis para leer a un archivo

- Archivo = `open(Ruta de archivo, permisos, encodeo)`
- Archivo.`close()`

```
archivo = open("C:\Users\usuario\Escritorio\texto.tx")
```

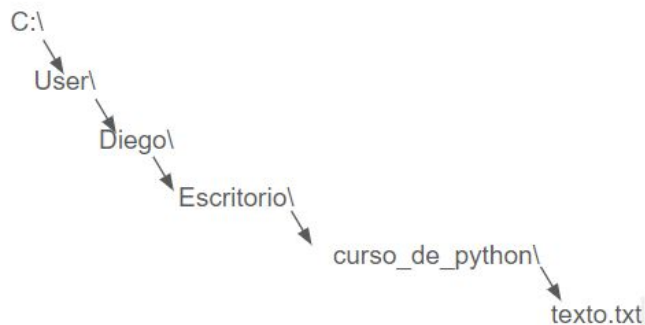
```
archivo.close()
```



Sintaxis para leer a un archivo

- Archivo = `open(Ruta de archivo, permisos, encodeo)`
- `Archivo.close()`
- `with open(Ruta de archivo) as Archivo:`

`with open(Ruta de archivo) as archivo:`




Sintaxis para leer a un archivo

- Archivo = `open(Ruta de archivo, permisos, encodeo)`
- `Archivo.close()`
- `with open(Ruta de archivo) as Archivo:`
- `Archivo.read()`

`with open(Ruta de archivo) as archivo:`

`print(archivo.read(4))`

>>> Hola



Hola Mundo
Hola Mundo2
Hola Mundo3

Sintaxis para leer a un archivo

- Archivo = `open(Ruta de archivo, permisos, encodeo)`
- `Archivo.close()`
- `with open(Ruta de archivo) as Archivo:`
- `Archivo.read()`

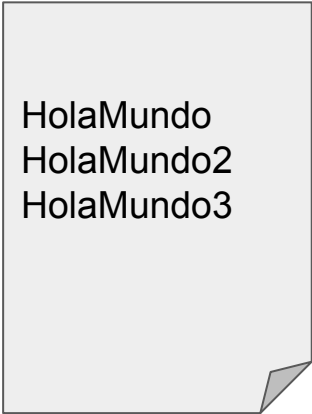
```
with open(Ruta de archivo,'r') as archivo:
```

```
    print(archivo.read(4))
```

```
    print(archivo.read(5))
```

```
>>> Hola
```

```
>>> Mundo
```



HolaMundo
HolaMundo2
HolaMundo3

Sintaxis para leer a un archivo

- Archivo = `open(Ruta de archivo, permisos, encodeo)`
- `Archivo.close()`
- `with open(Ruta de archivo) as Archivo:`
- `Archivo.read()`

```
with open(Ruta de archivo,'r') as archivo:
```

```
    print(archivo.readline())
```

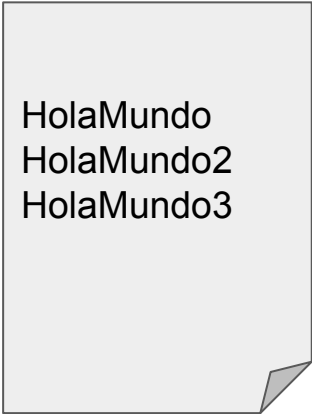
```
    print(archivo.readline())
```

```
    print(archivo.readline())
```

```
>>> HolaMundo
```

```
>>> HolaMundo2
```

```
>>> HolaMundo3
```



HolaMundo
HolaMundo2
HolaMundo3

Sintaxis para leer a un archivo

- Archivo = `open(Ruta de archivo, permisos, encodeo)`
- `Archivo.close()`
- `with open(Ruta de archivo) as Archivo:`
- `Archivo.read()`

`with open(Ruta de archivo,'r') as archivo:`

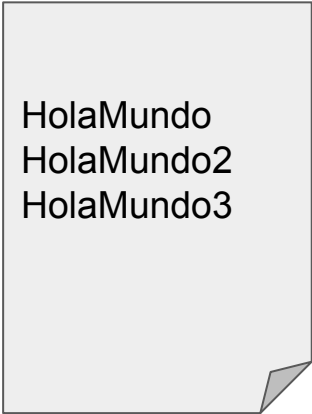
For linea in archivo:

`print(linea)`

`>>> HolaMundo`

`>>> HolaMundo2`

`>>> HolaMundo3`



HolaMundo
HolaMundo2
HolaMundo3

Sintaxis para leer a un archivo

- Archivo = `open(Ruta de archivo, permisos, encodeo)`
- `Archivo.close()`
- `with open(Ruta de archivo) as Archivo:`
- `Archivo.read()`

`with open(Ruta de archivo, 'r') as archivo:`

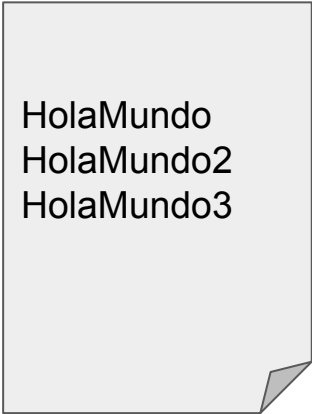
For linea in archivo:

`print(linea)`

>>> HolaMundo

>>> HolaMundo2

>>> HolaMundo3



HolaMundo
HolaMundo2
HolaMundo3

Sintaxis para escribir a un archivo

- Archivo = `open(Ruta de archivo, permisos, encodeo)`
- `Archivo.close()`
- `with open(Ruta de archivo) as Archivo:`
- `Archivo.read()`
- `Archivo.write(String)`

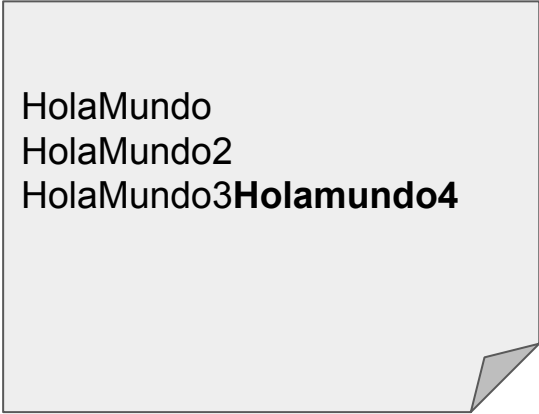
`with open(Ruta de archivo, 'w') as archivo:`

`archivo.write("Holamundo\n")`

`archivo.write("Holamundo2\n")`

`archivo.write("Holamundo3")`

`archivo.write("Holamundo4")`



HolaMundo
HolaMundo2
HolaMundo3Holamundo4

Sintaxis para escribir a un archivo

- Archivo = `open(Ruta de archivo, permisos, encodeo)`
- `Archivo.close()`
- `with open(Ruta de archivo) as Archivo:`
- `Archivo.read()`
- `Archivo.write(String)`
- Carácter especial para new line o nueva línea `/n`

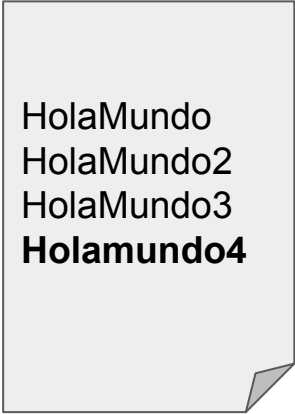
`with open(Ruta de archivo, 'w') as archivo:`

`archivo.write("Holamundo\n")`

`archivo.write("Holamundo2\n")`

`archivo.write("Holamundo3\n")`

`archivo.write("Holamundo4")`



HolaMundo
HolaMundo2
HolaMundo3
Holamundo4


Sintaxis para escribir un archivo

- Archivo = `open(Ruta de archivo, permisos, encodeo)`
- `Archivo.close()`
- `with open(Ruta de archivo) as Archivo:`
- `Archivo.read()`
- `Archivo.write(String)`
- Carácter especial para new line o nueva línea `/n`
- `Archivo.writelines(List[String,String, ..])`

```
lines = ['Holamundo\n','Holamundo2\n','Holamundo3\n']
```

```
with open(Ruta de archivo, 'w') as archivo:
```

```
    archivo.writelines(lines)
```



```
HolaMundo  
HolaMundo2  
HolaMundo3
```

Módulos

```
def entrada_datos(string_imprimir):  
    while True:  
        try:  
            valor = float(input(string_imprimir))  
            return valor  
        except ValueError:  
            print('Error: Por favor ingresa un valor  
numerico')
```

↓
Entrada de
datos

← Cálculos

```
operador = input('Operador: ')  
numero 1 = entrada_datos('Primer numero: ')  
numero_2 = entrada_datos('Segundo numero: ')  
  
if operador == '+':  
    resultado = numero 1 + numero 2  
    print('Resultado es:', resultado)  
elif operador == '-':  
    resultado = numero 1 - numero 2  
    print('Resultado es:', resultado)  
elif operador == '*':  
    resultado = numero 1 * numero 2  
    print('Resultado es:', resultado)  
elif operador == '/':  
    try:  
        resultado = numero 1 / numero 2  
        print('Resultado es:', resultado)  
    except Exception:  
        print('Error: No es posible dividir para zero')  
else:  
    print('Operador es invalido')
```


Dividiendo el programa


recibir_informacion.py

```
def entrada_datos(string_imprimir):  
    while True:  
        try:  
            valor = float(input(string_imprimir))  
            return valor  
        except ValueError:  
            print('Error: Por favor ingresa un valor  
numérico')
```

calculos.py

```
operador = input('Operador: ')  
numero_1 = entrada_datos('Primer numero: ')  
numero_2 = entrada_datos('Segundo numero: ')  
  
if operador == '+':  
    resultado = numero_1 + numero_2  
    print('Resultado es:', resultado)  
elif operador == '-':  
    resultado = numero_1 - numero_2  
    print('Resultado es:', resultado)  
elif operador == '*':  
    resultado = numero_1 * numero_2  
    print('Resultado es:', resultado)  
elif operador == '/':  
    try:  
        resultado = numero_1 / numero_2  
        print('Resultado es:', resultado)  
    except Exception:  
        print('Error: No es posible dividir para zero')  
else:  
    print('Operador es invalido')
```

Operador: +
Traceback (most recent call last):
File "calculos.py", line 2, in <module>
numero_1 = entrada_datos('Primer numero: ')
NameError: name 'entrada_datos' is not defined



Dividiendo el programa

calculos.py

```
from recibir_informacion import entrada_datos
```

```
operador = input('Operador: ')
numero_1 = entrada_datos('Primer numero: ')
numero_2 = entrada_datos('Segundo numero: ')
```

```
if operador == '+':
    resultado = numero_1 + numero_2
    print('Resultado es:', resultado)
    . . .
else:
    print('Operador es invalido')
```

calculos.py

```
import recibir_informacion
```

```
operador = input('Operador: ')
numero_1 = recibir_informacion.entrada_datos('Primer numero:')
numero_2 = recibir_informacion.entrada_datos('Segundo numero:')
```

```
if operador == '+':
    resultado = numero_1 + numero_2
    print('Resultado es:', resultado)
    . . .
else:
    print('Operador es invalido')
```

Final de la diapositiva



Ejemplo módulo fecha y módulo os

```
from datetime import date
```

```
print("El día de hoy es ", date.today())
```

```
>> El día de hoy es 2023-04-04
```

```
import os
```

```
os.mkdir('mi_nuevo_directorio')
```

Final de la diapositiva



Conclusión

- Hay varias opciones de almacenamiento de datos.
- Las opciones de almacenamiento varían con respecto a la velocidad.
- Algunas opciones de almacenamiento guardan los datos incluso cuando la computadora está apagada.
- Un archivo guarda información en memoria externa.
- La ruta de un archivo define su posición dentro del sistema de ficheros.
- La ruta de ficheros define una jerarquía del sistema.
- Sintaxis para abrir archivos en python indicando si queremos abrirlos como lectura o escritura.
- Un módulo identifica instrucciones importadas dentro de otro programa.