

Introducción a la programación con Python

Manejo de errores, Archivos, Módulos

Alexis Rodríguez

Marcel Morán C

Esquema

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

¿Qué es manejo de errores?

```
def division_ejemplo(numero1, numero2):  
    resultado = numero1 / numero2  
    print('Resultado de la division es:', resultado)
```

`division_ejemplo(4, 2)`



Resultado de la division es: 2.0

`division_ejemplo(4, 0)`

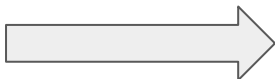


```
ZeroDivisionError                                Traceback (most recent call last)  
~\AppData\Local\Temp\ipykernel_24784\1793463972.py in <module>  
3     print('Resultado de la division es:', resultado)  
4  
----> 5 division_ejemplo(2, 0)  
~\AppData\Local\Temp\ipykernel_24784\1793463972.py in division_ejemplo(numero1, numero2)  
1     def division_ejemplo(numero1, numero2):  
----> 2         resultado = numero1 / numero2  
3         print('Resultado de la division es:', resultado)  
4  
5     division_ejemplo(4, 0)  
ZeroDivisionError: division by zero
```

Sintaxis de manejo de errores

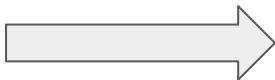
```
def division_ejemplo(numero1, numero2):  
    try:  
        resultado = numero1 / numero2  
        print('Resultado es:', resultado)  
    except ZeroDivisionError:  
        print('Error: No es posible dividir para zero')
```

division_ejemplo(4, 2)



Resultado de la division es: 2.0

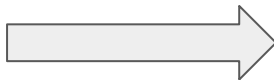
division_ejemplo(4, 0)



Error: No es posible dividir para zero

Tipos de excepciones

```
mistring = '45'  
minumero = int(mistring)  
print('mi numero es:', minumero)
```



mi numero es: 45

```
mi string = 'hello'  
mi numero = int(mi string)  
print('mi numero es:', mi_numero)
```



ValueError

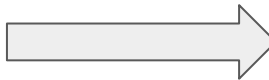
Traceback (most recent call last)

~\AppData\Local\Temp\ipykernel_24784\2402285655
.py in <module>
1 mi string = 'hello'
----> 2 mi_numero = int(mi_string)
3 print('mi numero es:', mi_numero)

ValueError: invalid literal for int() with
base 10: 'hello'

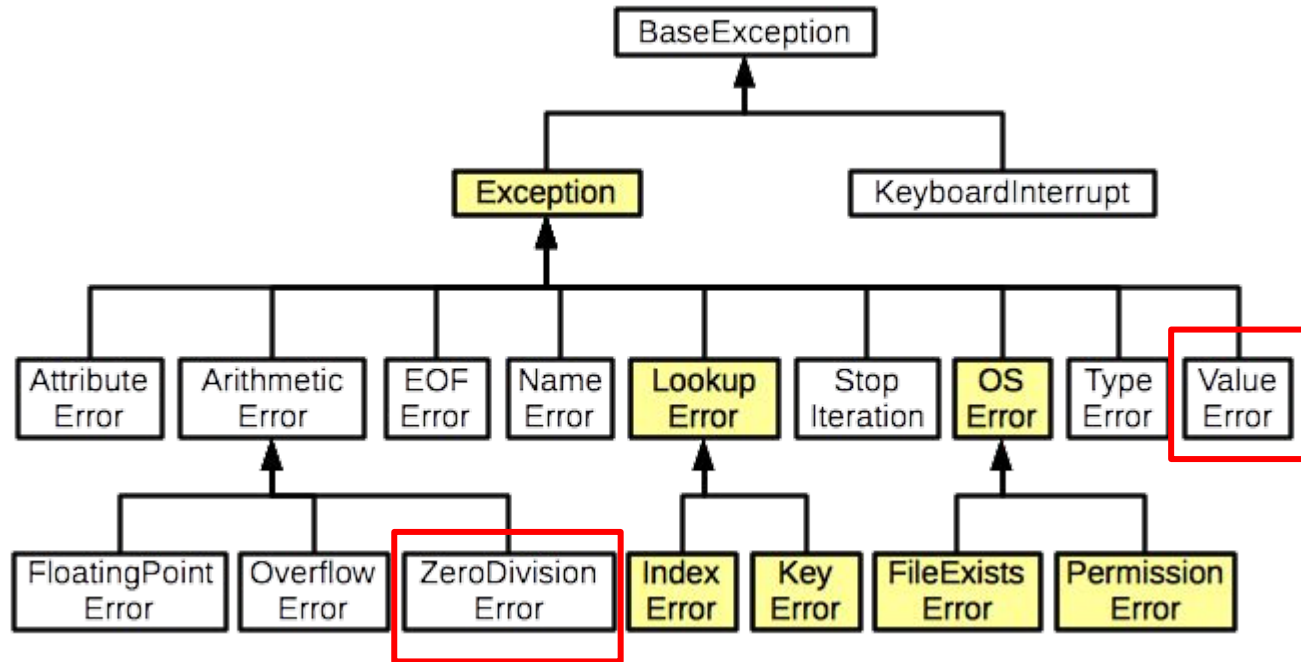
Tipos de excepciones

```
mi_string = 'hello'
try:
    mi_numero = int(mi_string)
    print('mi numero es:', mi_numero)
except ValueError:
    print('Error: Valor invalido', "" + mi_string + "")
```



Error: Valor invalido 'hello'

Tipos de excepciones



Tipos de excepciones

```
try:
```

```
...
```

```
except PrimeraExcepcion:  
    manejar_primera()
```

```
except SegundaExcepcion:  
    manejar_segunda()
```

```
except (TerceraExcepcion, CuartaExcepcion, QuintaExcepcion) as e:  
    manejar_3ra_4ta_5ta()
```

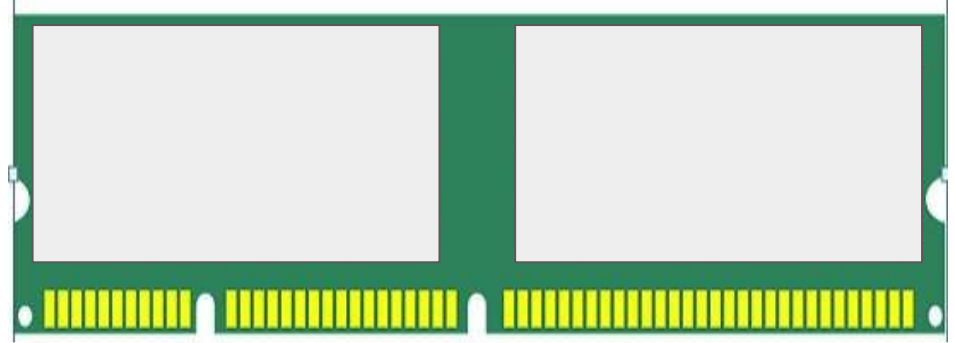
```
except Exception:  
    manejar_todo_lo_demas()
```

← Incluye a todos los
tipos de excepciones

Guardando datos

- Datos en RAM (tipo interno de memoria)
- ¿Qué pasa cuando acabamos nuestro programa?

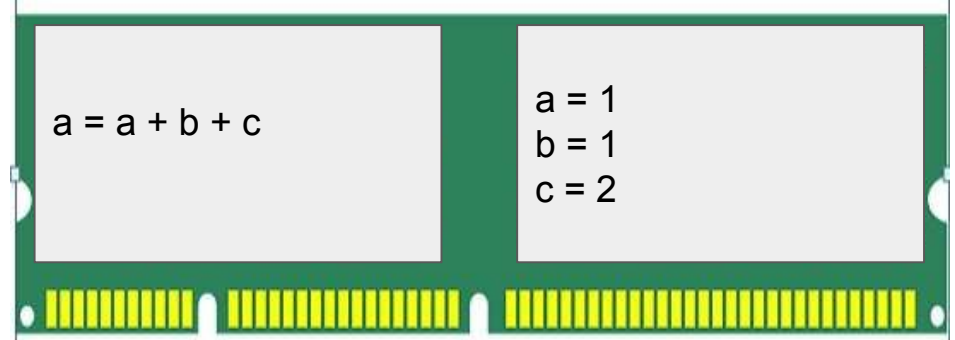
```
a = 1  
b = 1  
c = 2  
a = a + b + c
```



Guardando datos

- Datos en RAM (tipo interno de memoria)
- ¿Qué pasa cuando acabamos nuestro programa?

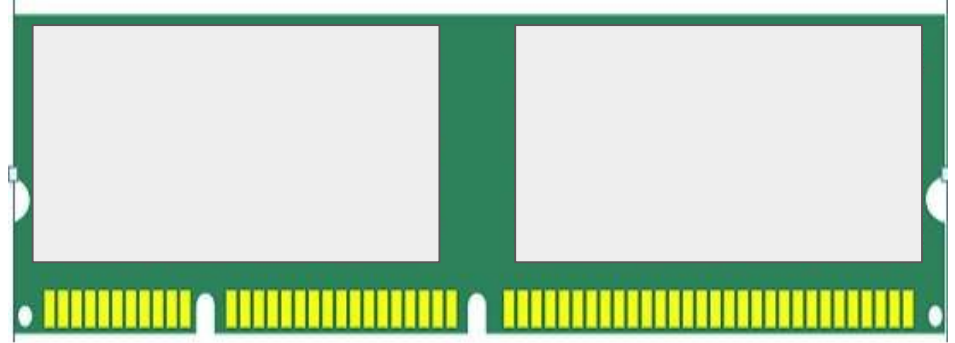
```
a = 1  
b = 1  
c = 2  
a = a + b + c
```



Guardando datos

- Datos en RAM (tipo interno de memoria)
- ¿Qué pasa cuando acabamos nuestro programa?
- Datos no persisten al terminar la ejecución del programa

```
a = 1  
b = 1  
c = 2  
a = a + b + c
```



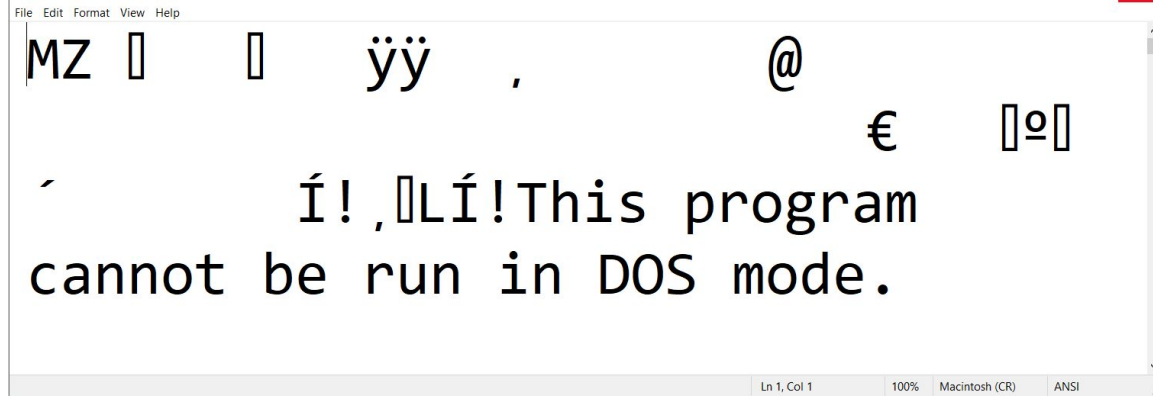
Guardando datos

- Una vez 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**)



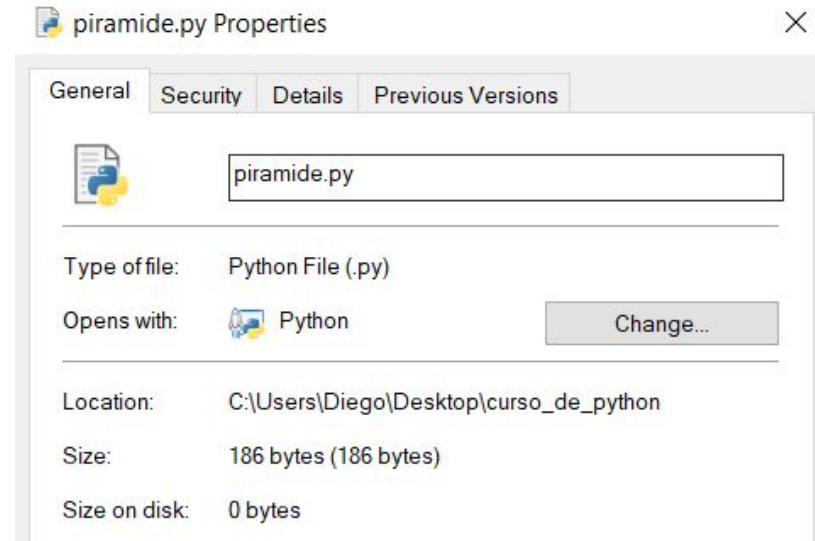
The image shows a screenshot of a text editor window, likely a DOS-style application. The window has a menu bar with 'File', 'Edit', 'Format', 'View', and 'Help'. The text content is as follows:

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

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

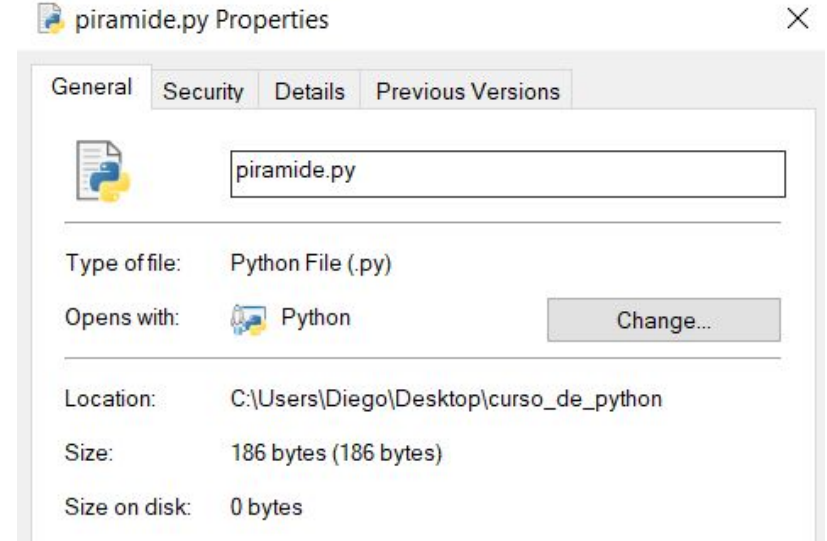
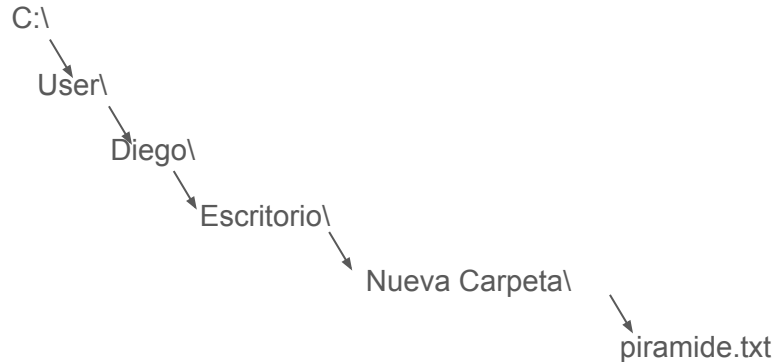
Ruta de archivos

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



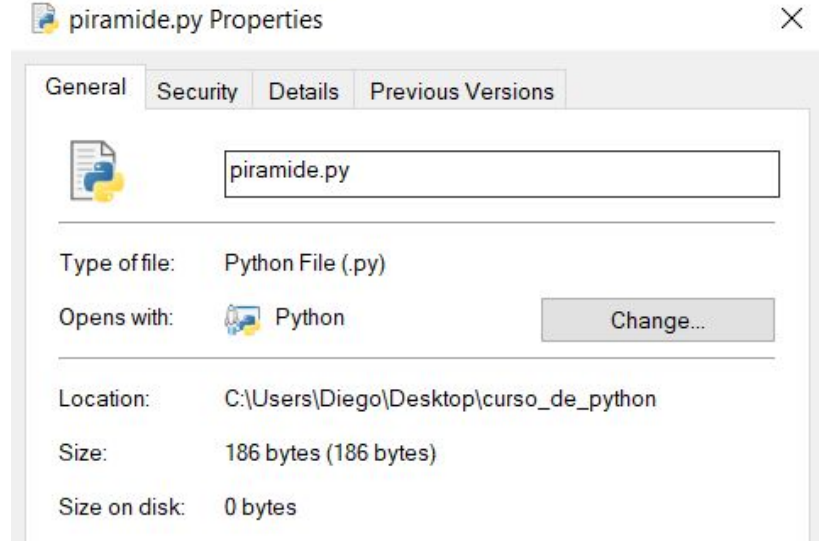
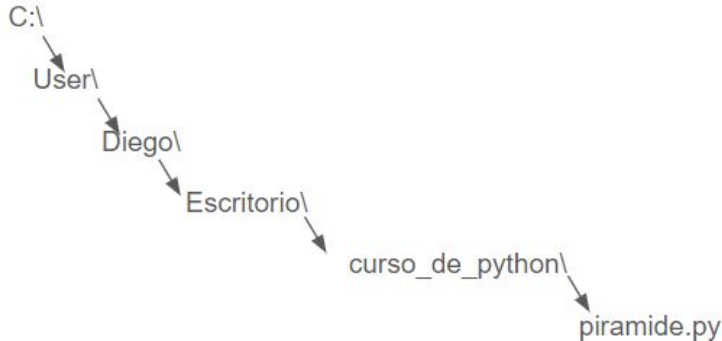
Ruta de archivos

- 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:\

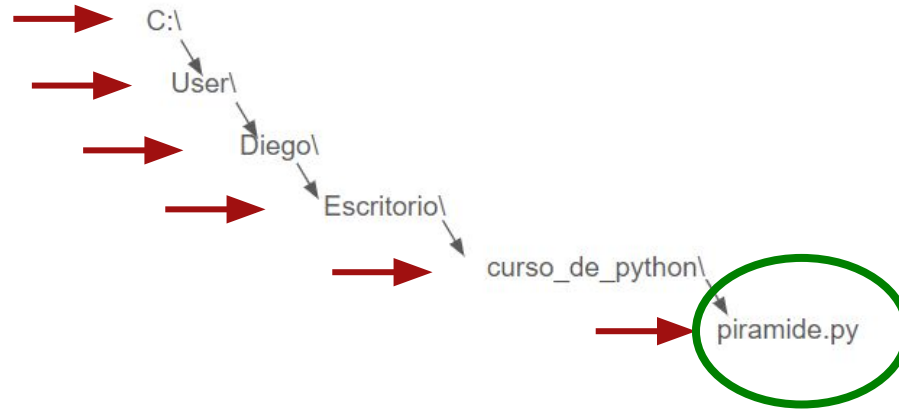


Ruta de archivos

- 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

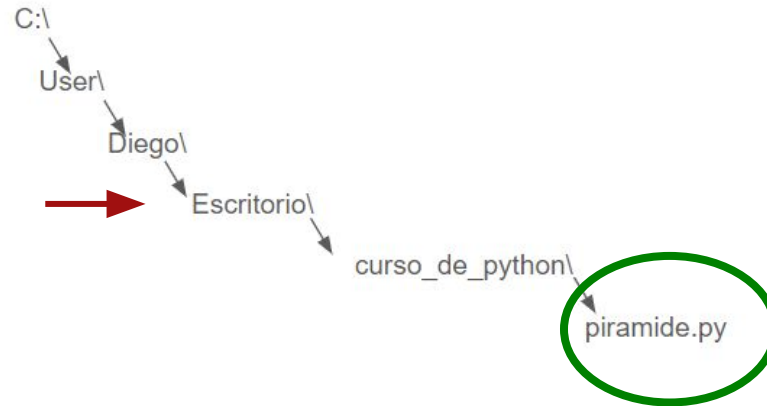
Ruta de archivos

- 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**

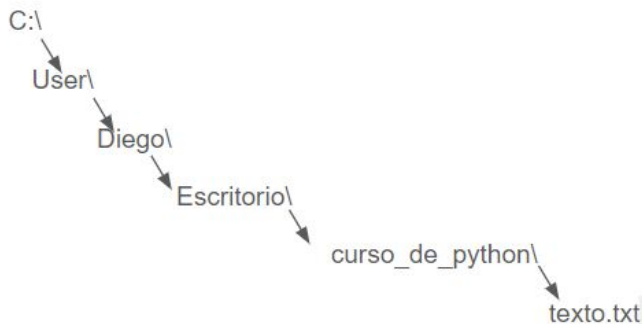


Sintaxis para leer un archivo

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

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

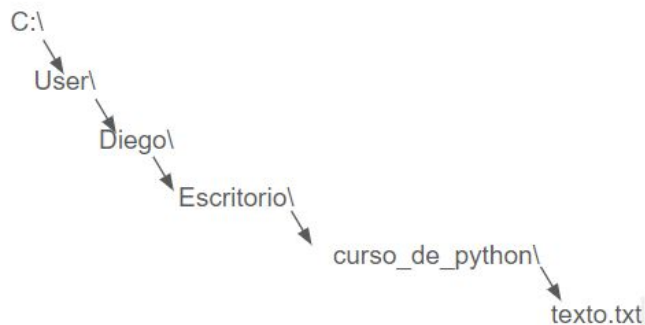
```
archivo.close()
```



Sintaxis para leer 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:`



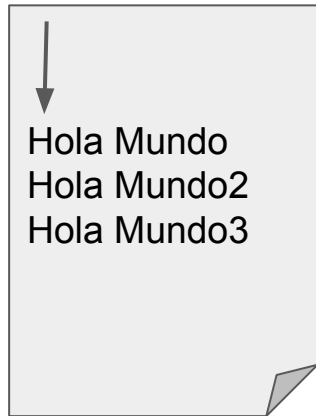
Sintaxis para leer un archivo

- 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



Sintaxis para leer un archivo

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

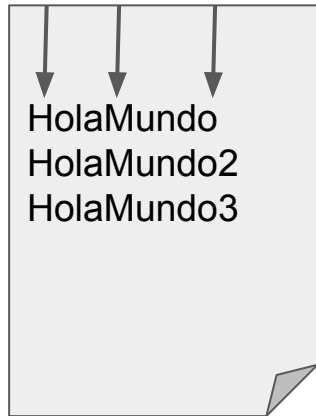
`with open(Ruta de archivo, ") as archivo:`

—→ `print(archivo.read(4))`

—→ `print(archivo.read(5))`

>>> Hola

>>> Mundo



Sintaxis para leer 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.readline())
```

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

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


```
>>> HolaMundo
```

```
>>>
```

```
>>> HolaMundo2
```

```
>>>
```

```
>>> HolaMundo3
```



```
HolaMundo\nHolaMundo2\nHolaMundo3\n
```


Sintaxis para leer 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(),end="")
```


```
    print(archivo.readline(),end="")
```

```
    print(archivo.readline(),end="")
```

```
>>> HolaMundo
```

```
>>> HolaMundo2
```

```
>>> HolaMundo3
```



```
HolaMundo\nHolaMundo2\nHolaMundo3
```

Sintaxis para leer 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:
```

```
    lineas = archivo.readlines()
```


```
    for linea in lineas:
```

```
        print(linea, end="")
```

```
>>> HolaMundo
```

```
>>> HolaMundo2
```

```
>>> HolaMundo3
```



```
HolaMundo\nHolaMundo2\nHolaMundo3\n
```

Sintaxis para leer 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:`

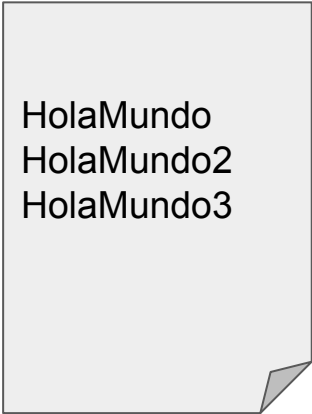
`for linea in archivo:`

`print(linea, end="")`

`>>> 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)`

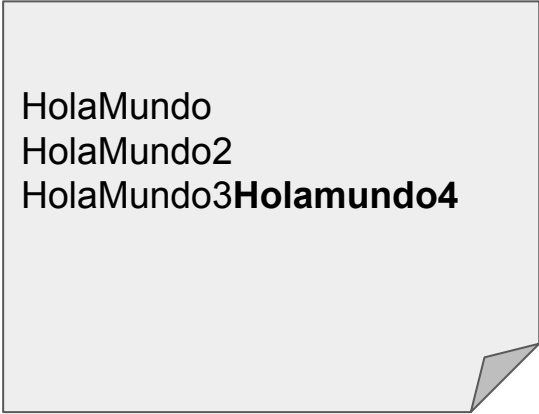
`with open(Ruta de archivo, 'w') as archivo:`

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

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

`archivo.write('Holamundo3')`

`archivo.write('Holamundo4')`



HolaMundo
HolaMundo2
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`

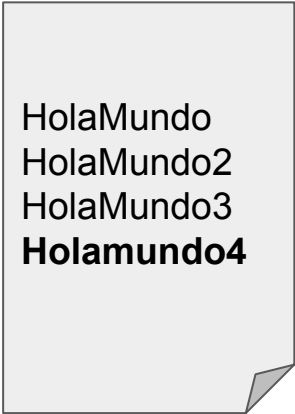
`with open(Ruta de archivo,'w') as archivo:`

`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
```

¿Qué es un módulo?

- Instrucciones guardadas en un archivo
- Instrucciones son importadas dentro de otro programa
- La collection de estos módulos se conocen librerías
- **OS y datetime son un ejemplo de estos**

```
os.makedirs('C:\\Users\\usuario\\Escritorio\\Nueva Carpeta')
```



¿Qué es un módulo?

- Instrucciones guardadas en un archivo
- Instrucciones son importadas dentro de otro programa
- La collection de estos módulos se conocen librerías
- **OS y datetime son un ejemplo de estos**

```
from datetime import date
```

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

```
>>> El dia de hoy es 2022-03-11
```


Conclusión

- Cómo manejar errores
- Sintaxis de excepciones (try, except) y los tipos que existen
- 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.

Retroalimentación

- Para retroalimentación dirigirse al siguiente enlace <https://forms.gle/UEtzxHteT6NKYipj8> .
- Déjanos saber qué podemos hacer para mejorar el curso

