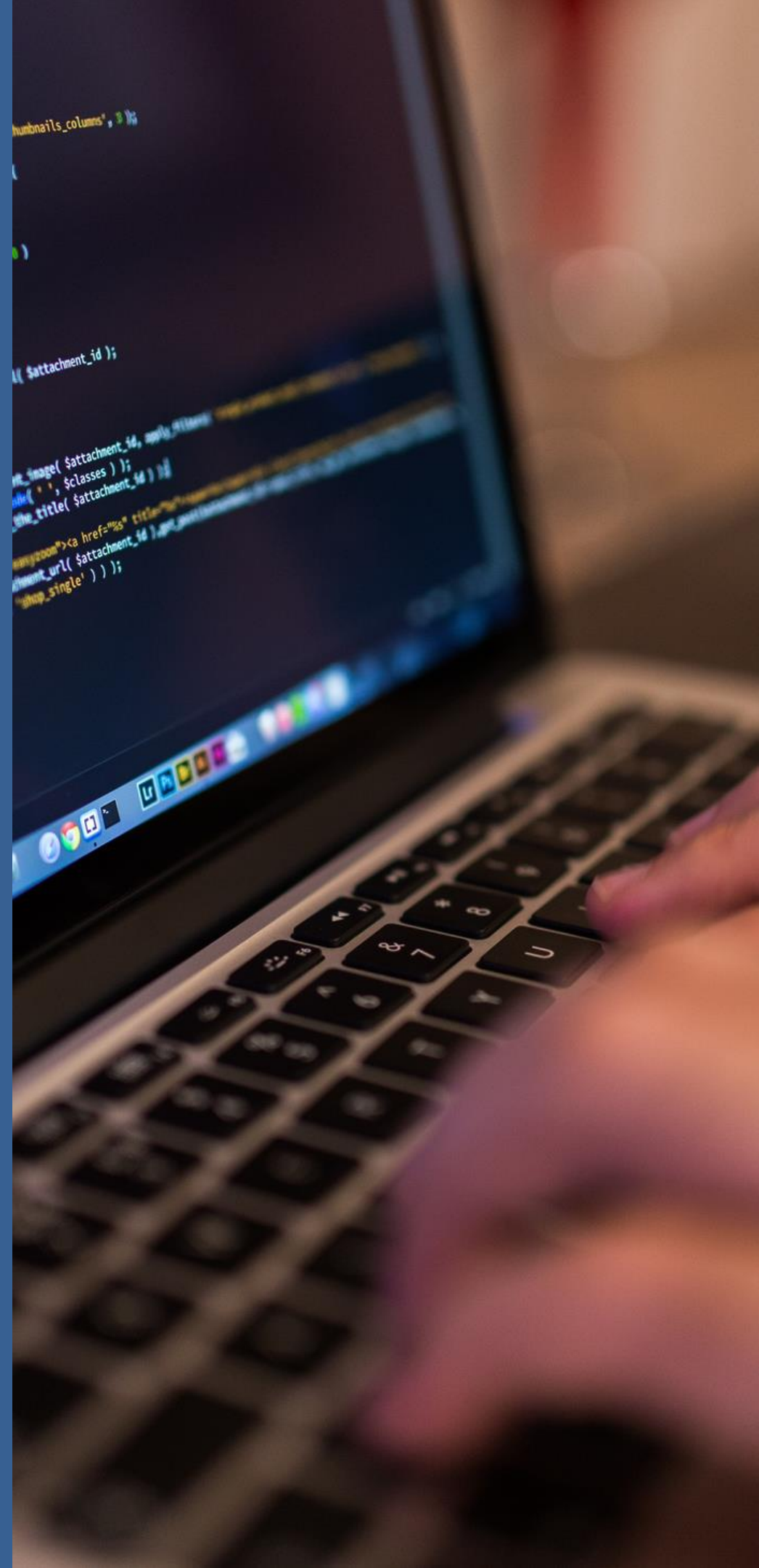


ESTRUCTURA DE DATOS Y PROGRAMACIÓN

ITBA

Prof. Ninfa Esperanza Delgado Bolivar
ndelgado@itba.edu.ar
ninfadelgado@hotmail.com





EN PYTHON TODO ES UN OBJETO

Temario de la Clase



- ✓ Qué es un Archivo?
- ✓ Tipos de archivos
- ✓ Cómo Leer / Escribir en los archivos ?



EN PYTHON TODO ES UN OBJETO

TEXTO

BINARIOS



CSV

JPG



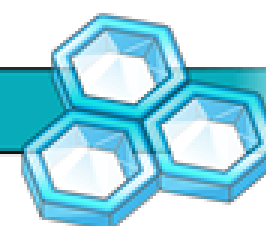
Cómo se ve un Archivo

Texto: “ejemplo1.py”

Binario: “8.Archivos – Actividades.pdf”

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	
00000000	23	50	72	69	6d	65	72	6f	20	73	65	20	65	73	63	72	#Primero se escr
00000010	69	62	65	6e	20	6c	6f	73	20	69	6d	70	6f	72	74	0d	iben los import.
00000020	0a	66	72	6f	6d	20	72	61	6e	64	6f	6d	20	69	6d	70	.from random imp
00000030	6f	72	74	20	2a	0d	0a	23	4c	75	65	67	6f	20	73	65	ort *..#Luego se
00000040	20	65	73	63	72	69	62	65	6e	20	6c	61	73	20	64	65	escriben las de
00000050	66	69	6e	69	63	69	6f	6e	65	73	20	64	65	20	66	75	finiciones de fu
00000060	6e	63	69	6f	6e	65	73	20	70	72	6f	70	69	61	73	0d	nciones propias.
00000070	0a	64	65	66	20	65	73	50	72	69	6d	6f	28	6e	29	3a	.def esPrimo(n):
00000080	0d	0a	20	20	20	20	70	72	69	6d	6f	20	3d	20	54	72	.. primo = Tr
00000090	75	65	0d	0a	20	20	20	20	69	66	20	6e	20	3e	20	32	ue.. if n > 2
000000a0	3a	0d	0a	20	20	20	20	20	20	20	20	64	69	76	20	3d	:.. div =
000000b0	20	32	0d	0a	20	20	20	20	20	20	20	20	77	68	69	6c	2.. while
000000c0	65	20	64	69	76	20	3c	20	6e	20	61	6e	64	20	70	72	e div < n and pr

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	
00000000	25	50	44	46	2d	31	2e	37	0d	0a	25	b5	b5	b5	b5	0d	%PDF-1.7...%µµµµ.
00000010	0a	31	20	30	20	6f	62	6a	0d	0a	3c	3c	2f	54	79	70	.1 0 obj...<</Typ
00000020	65	2f	43	61	74	61	6c	6f	67	2f	50	61	67	65	73	20	e/Catalog/Pages
00000030	32	20	30	20	52	2f	4c	61	6e	67	28	65	73	2d	4d	58	2 0 R/Lang(es-MX
00000040	29	20	2f	4d	65	74	61	64	61	74	61	20	33	34	20	30) /Metadata 34 0
00000050	20	52	2f	56	69	65	77	65	72	50	72	65	66	65	72	65	R/ViewerPrefere
00000060	6e	63	65	73	20	33	35	20	30	20	52	3e	3e	0d	0a	65	nces 35 0 R>>..e
00000070	6e	64	6f	62	6a	0d	0a	32	20	30	20	6f	62	6a	0d	0a	ndobj..2 0 obj..
00000080	3c	3c	2f	54	79	70	65	2f	50	61	67	65	73	2f	43	6f	<</Type/Pages/Co
00000090	75	6e	74	20	31	2f	4b	69	64	73	5b	20	33	20	30	20	unt 1/Kids[3 0
000000a0	52	5d	20	3e	3e	0d	0a	65	6e	64	6f	62	6a	0d	0a	33	R] >>..endobj..3
000000b0	20	30	20	6f	62	6a	0d	0a	3c	3c	2f	54	79	70	65	2f	0 obj...<</Type/
000000c0	50	61	67	65	2f	50	61	72	65	6e	74	20	32	20	30	20	Page/Parent 2 0





Archivos en Python

•ARCHIVOS (FILES)

- De texto
- Tipo CSV (comma separator view)
- binarios

•RUTAS (PATHS)

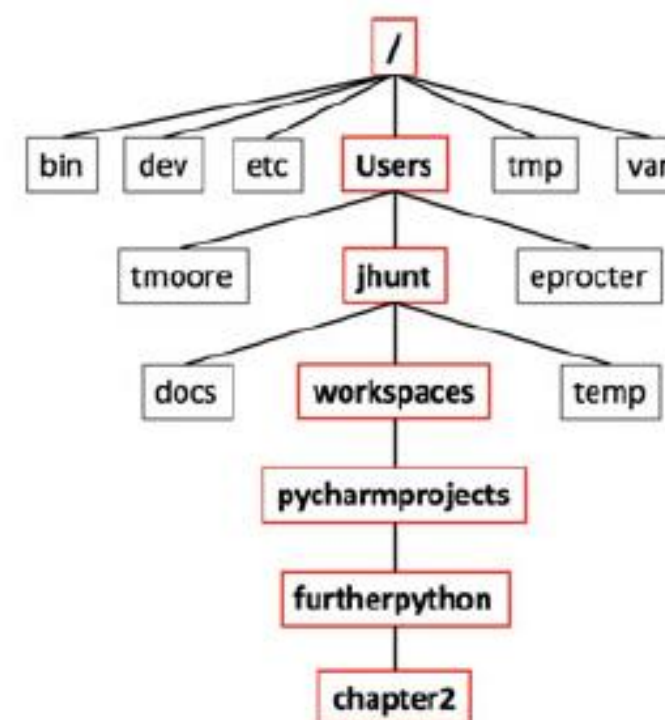
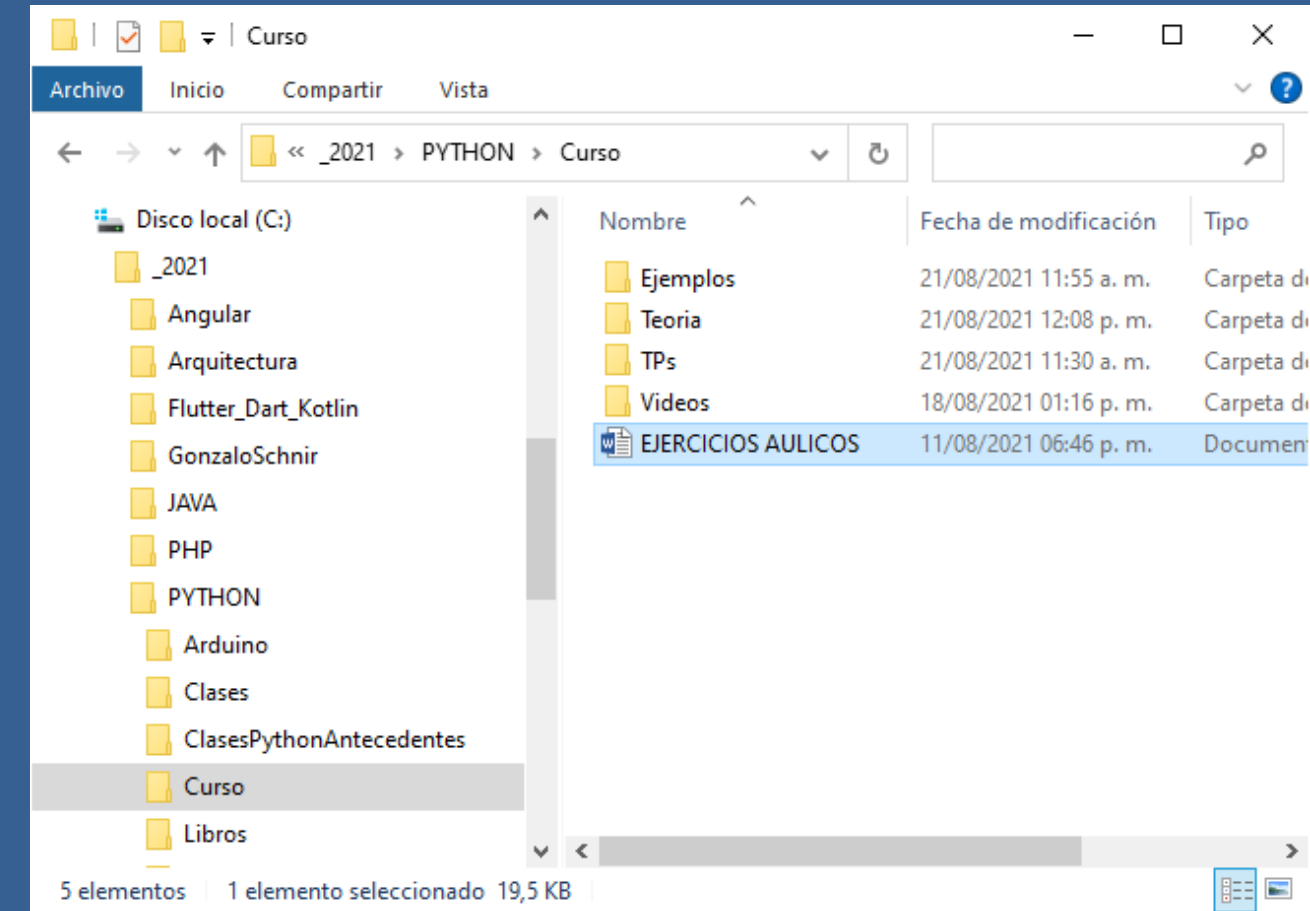
- Rutas Absolutas y relativas
 - Directorios
 - Raiz
 - subdirectorios

• UNIDADES DE MONTAJE (IO=input/output)

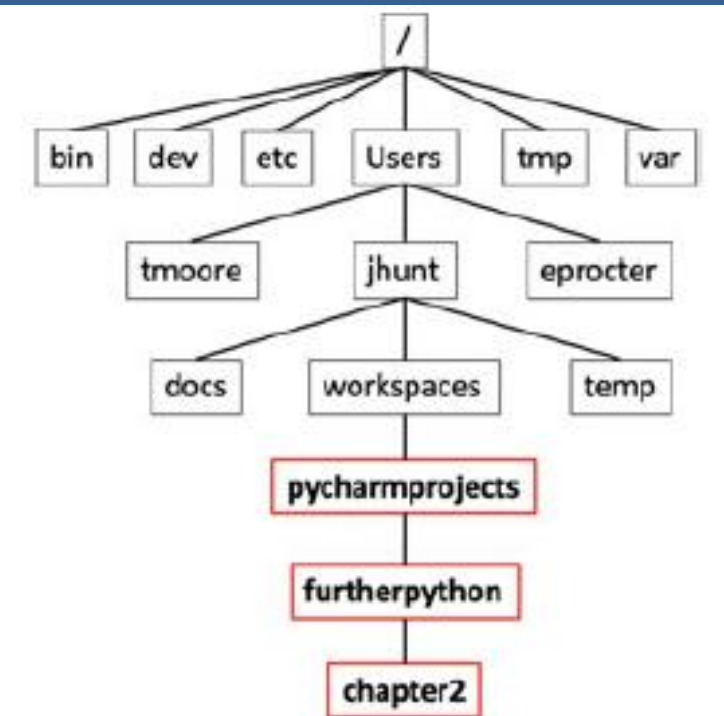
- Disco rigido
- usb
- nube



Archivos en Python



Path: /Users/jhunt/workspaces/pycharmprojects/furtherpython/chapter2



Relative path: pycharmprojects/furtherpython/chapter2



Acciones en los Archivos

- ✓ Abrir, Leer/Escribir, Cerrar FILES
- ✓ En la apertura tenemos las siguientes operaciones
 - + **file= open(nombreArchivo, modoAcceso, buffer)**
 - + **Buffer:**
 - valor 0 => no se hace buffering
 - valor 1 => el buffering se realiza mientras se accede al archivo



EN PYTHON TODO ES UN OBJETO

Archivos TXT



- Un archivo de texto es un archivo que sólo contiene caracteres ASCII.
- Se puede crear, abrir y editar con cualquier editor de texto.
- Acciones posibles:
 - Abrir para lectura
 - Leer símbolo a símbolo
 - Leer línea por línea
 - Crear (abrir para escritura)
 - Escribir un símbolo
 - Escribir una línea
 - Cerrar



Codigo	Detalle del modo de acceso
w	Write : modo de escritura. Se usa para crear un archivo nuevo. Cualquier archivo existente con el mismo nombres, será BORRADO y reemplazado por uno nuevo
r	Read : modo de lectura. Se abre solo para leerlo
a	Append : modo de agregar datos Se usa para actualizar el archivo con nueva información, NO PISA la información anterior, agrega al final del archivo lo nuevo



Codigo	Detalle del modo de acceso
r	Lectura de un archivo. El puntero se localiza al principio
rb	Lectura de un archivo binario. El puntero se localiza al principio
r+	Lectura y escritura de un archivo. El puntero se localiza al principio
rb+	Lectura y escritura de un archivo binario . El puntero se localiza al principio
w	Escritura de un archivo. Sobrescribe el archivo si ya existe, y si no existe crea uno nuevo
wb	Escritura de un archivo binario. El puntero se localiza al principio Sobrescribe el archivo si ya existe, y si no existe crea uno nuevo
w+	Lectura y escritura de un archivo. El puntero se localiza al principio Sobrescribe el archivo si ya existe, y si no existe crea uno nuevo
wb+	Lectura y escritura de un archivo binario . El puntero se localiza al principio Sobrescribe el archivo si ya existe, y si no existe crea uno nuevo
a	Escritura de un archivo para agregar. El puntero se localiza al final, agrega algo a lo que ya existe, NO SOBRESCRIBE. Si no existe el archivo, crea uno nuevo
ab	Escritura de un archivo binario. El puntero se localiza al final, agrega algo a lo que ya existe, NO SOBRESCRIBE. Si no existe el archivo, crea uno nuevo
a+	Escritura/lectura de un archivo. El puntero se localiza al final, agrega algo a lo que ya existe, NO SOBRESCRIBE. Si no existe el archivo, crea uno nuevo
ab+	Escritura/lectura de un archivo binario. El puntero se localiza al final, agrega algo a lo que ya existe, NO SOBRESCRIBE. Si no existe el archivo, crea uno nuevo



A Los Ejemplos



EN PYTHON TODO ES UN OBJETO

Archivos CSV



- Comma separator view
- `import csv`
 - `csv.reader`
 - `csv.writer`
 - `csv.writerow`
 - `csv.writerows`
 - `csv.DictWriter`, `csv.DictReader`
se usa cuando la 1era fila contiene el nombre de las columnas



A Los Ejemplos



Manejo Excepciones en Python

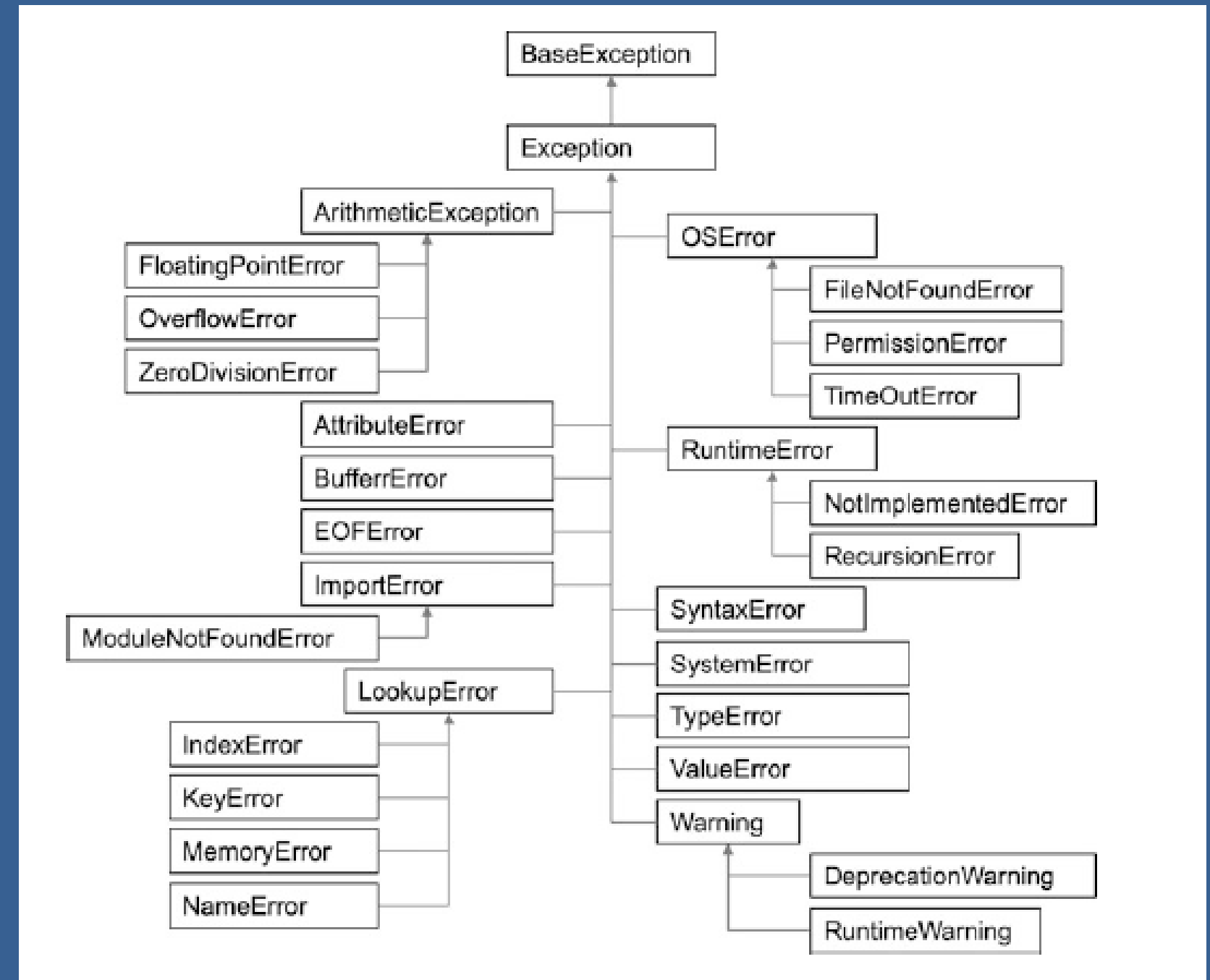
- ERROR

- Pueden ser asociados con temas funcionales

-

- EXCEPTION

- Pueden ser asociados con temas aritméticos, por ejemplo

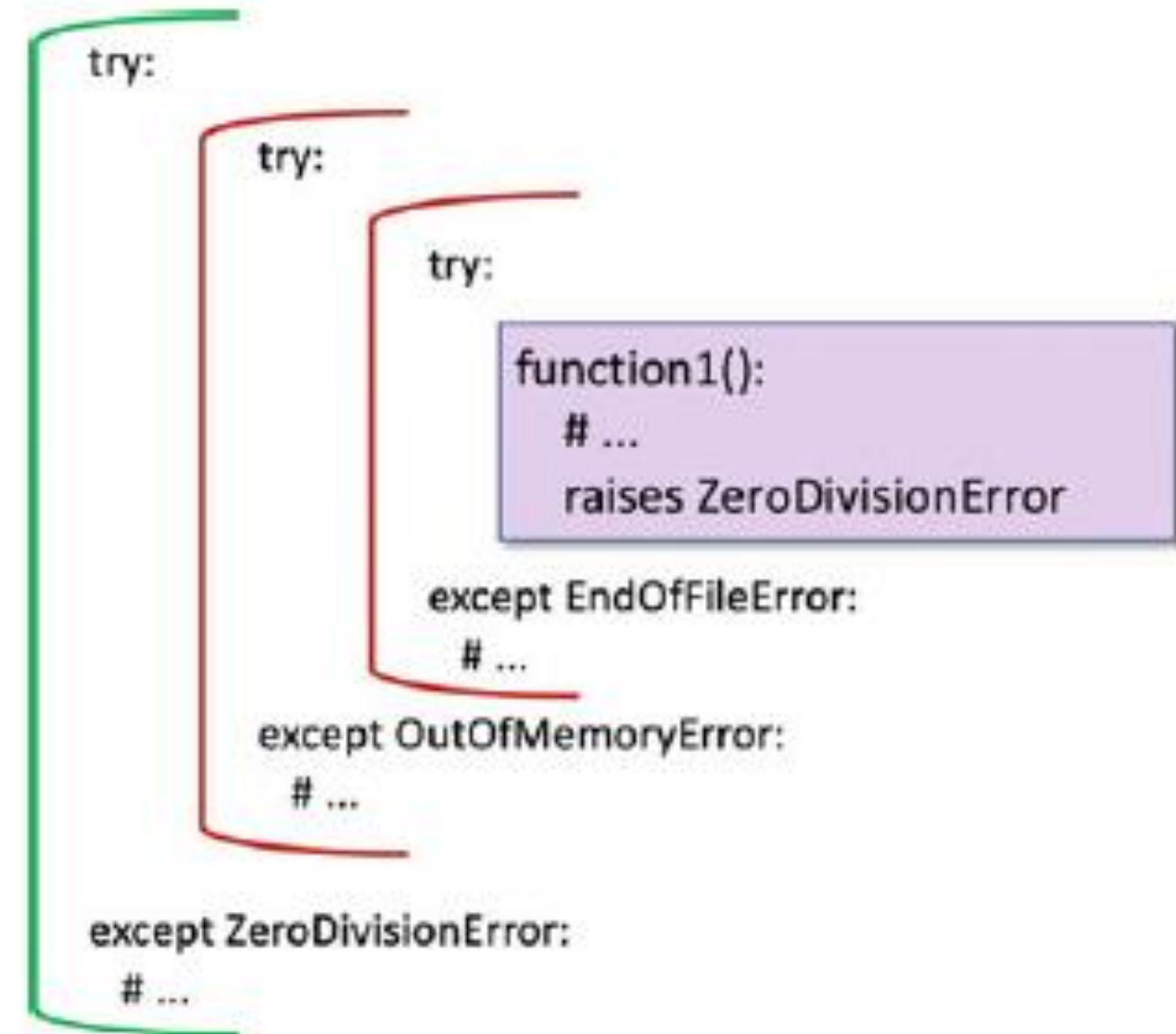


try:

.. Código a ejecutar

except ..tipoException:

... código a llamar si ocurre el error





A Los Ejemplos