

# Curso Python 2025 Día 5: Excepciones e iteración



# Hoy:

- ✓ Repaso: Slicing y mutabilidad
- ✓ Excepciones
- ✓ Iteradores
- ✓ Loops



# Repaso: Slicing

Extraer un “rango” de elementos de una secuencia.  
Indexar pero en vez de **int** utilizamos:

- Notación → **start:stop:step**

`my_seq[start:stop:step]`

- Objeto → **slice(start, stop, step)**

`my_seq[slice(start, stop, step)]`

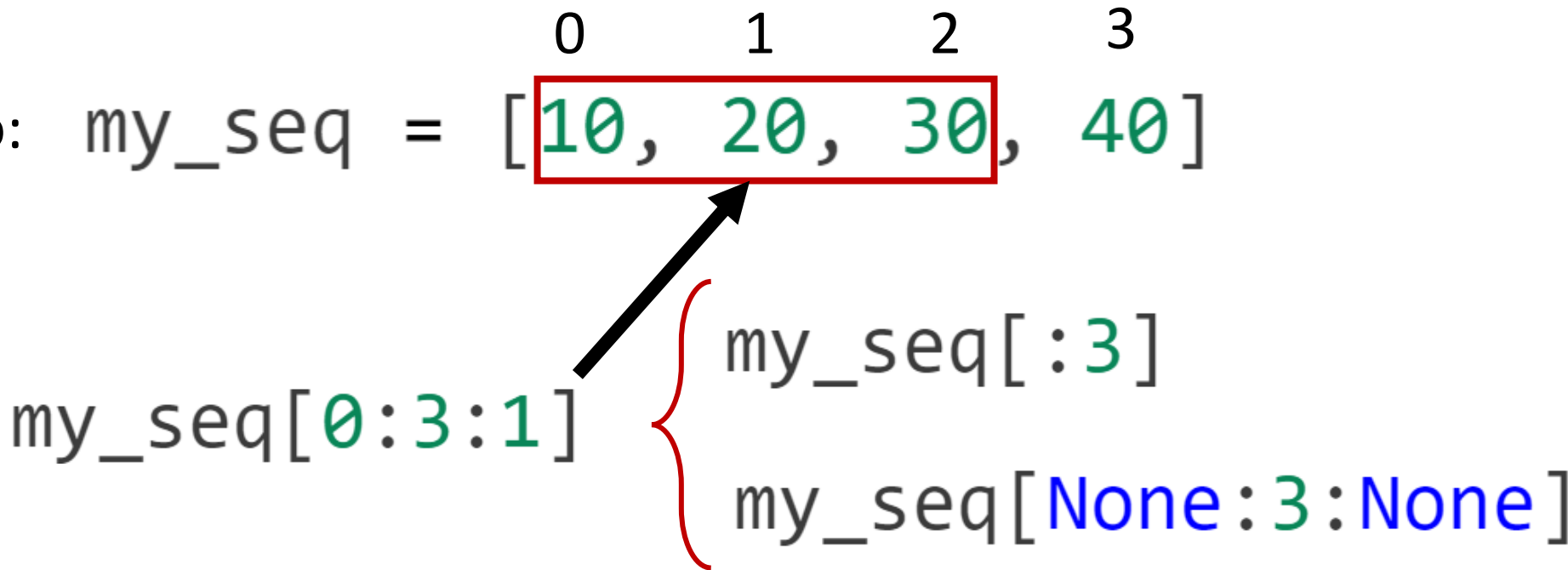
stop no incluido



Recordatorio: `my_sequence[x] == my_sequence.__getitem__(x)`

# Repaso: Slicing

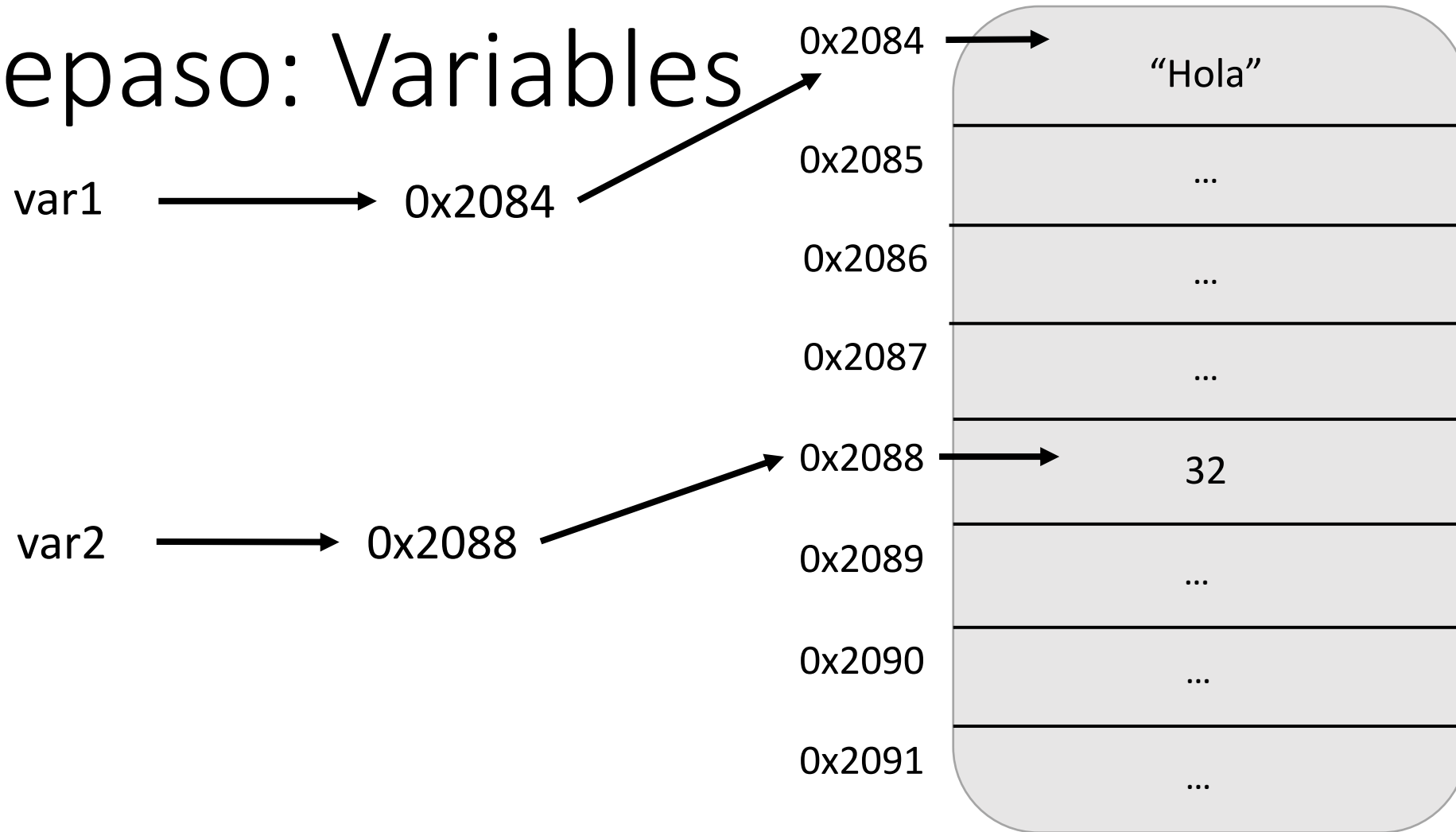
Ejemplo: `my_seq = [10, 20, 30, 40]`



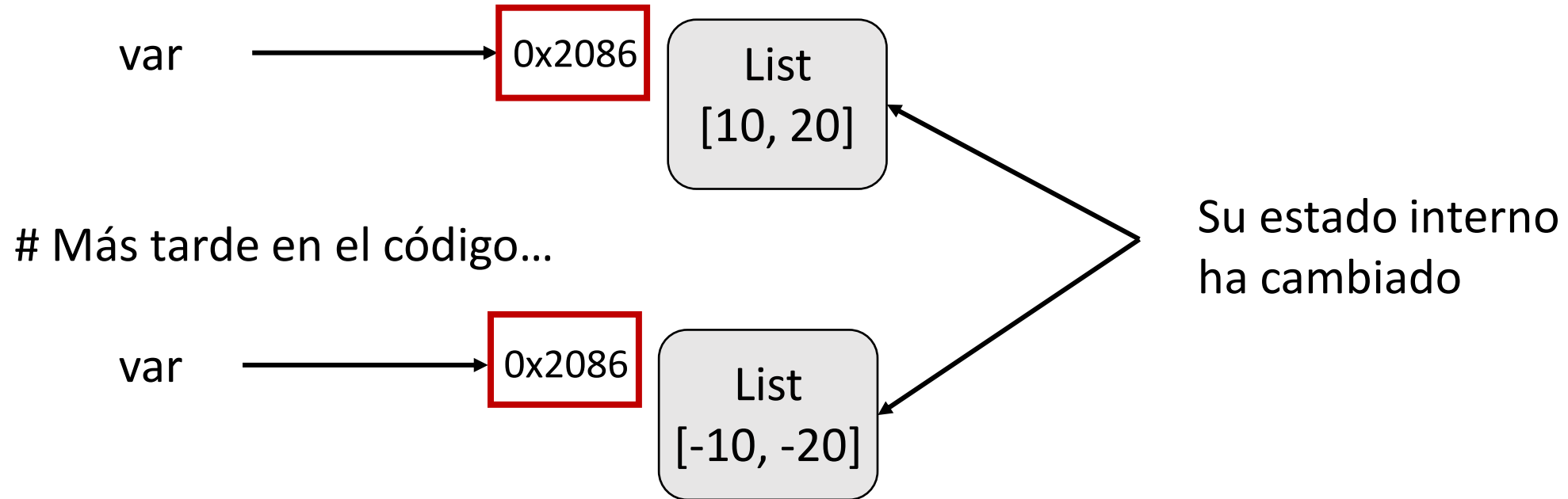
`my_seq[0:3:1]`  $\left\{ \begin{array}{l} \text{my\_seq[:3]} \\ \text{my\_seq[None:3:None]} \end{array} \right.$

Nota: Si no especificamos o damos valor **None** a start, stop o step por defecto toman valores 0, `len(my_seq)`, 1

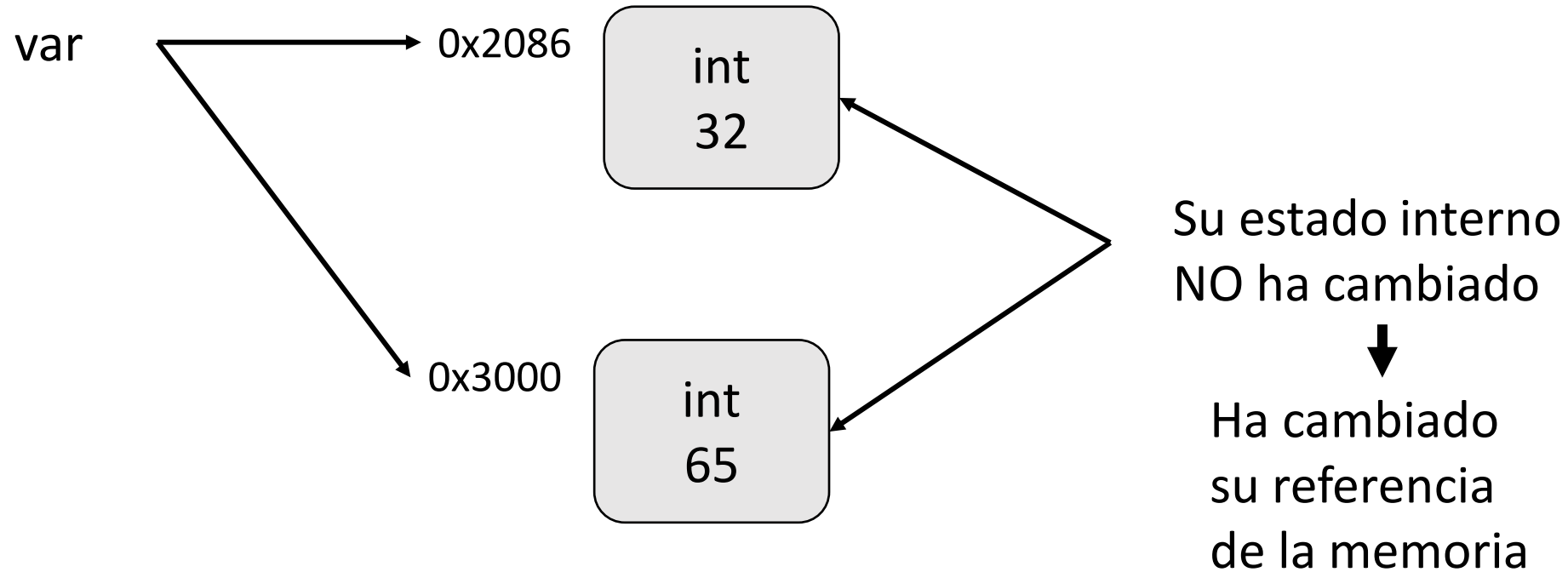
# Repaso: Variables



# Repaso: Objetos mutables



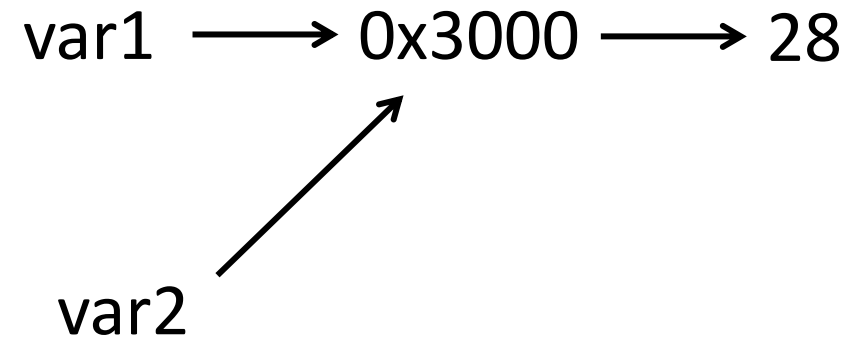
# Repaso: Objetos inmutables



Cuando realizamos una operación entre int obtenemos un nuevo objeto **int** !!!

# Repaso: Shared references

- Es posible que un objeto inmutable contenga objetos mutables
- Cuando asignamos a una variable otra se comparte la referencia **!!!**
- Si queremos una copia **copy** y **deepcopy**





# Excepciones: Concepto

Una excepción es un evento que **normalmente** ocurre cuando Python se encuentra con un comportamiento inesperado

- Hace que el programa termine de manera abrupta si no se gestiona
- Si no la gestionamos el programa termina

IndexError



[1, 2][2]

1 / 0

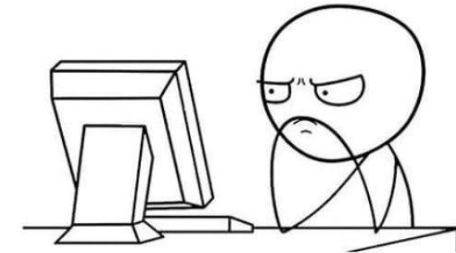


ZeroDivisionError

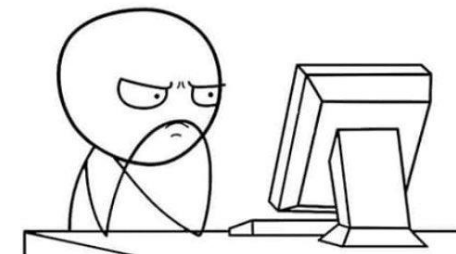
# Excepciones: Terminología

- **Exception** es un tipo de objeto en Python
- Se puede “arrojar” **raise Exception**
- Se puede gestionar, lo que llamamos **exception handling**

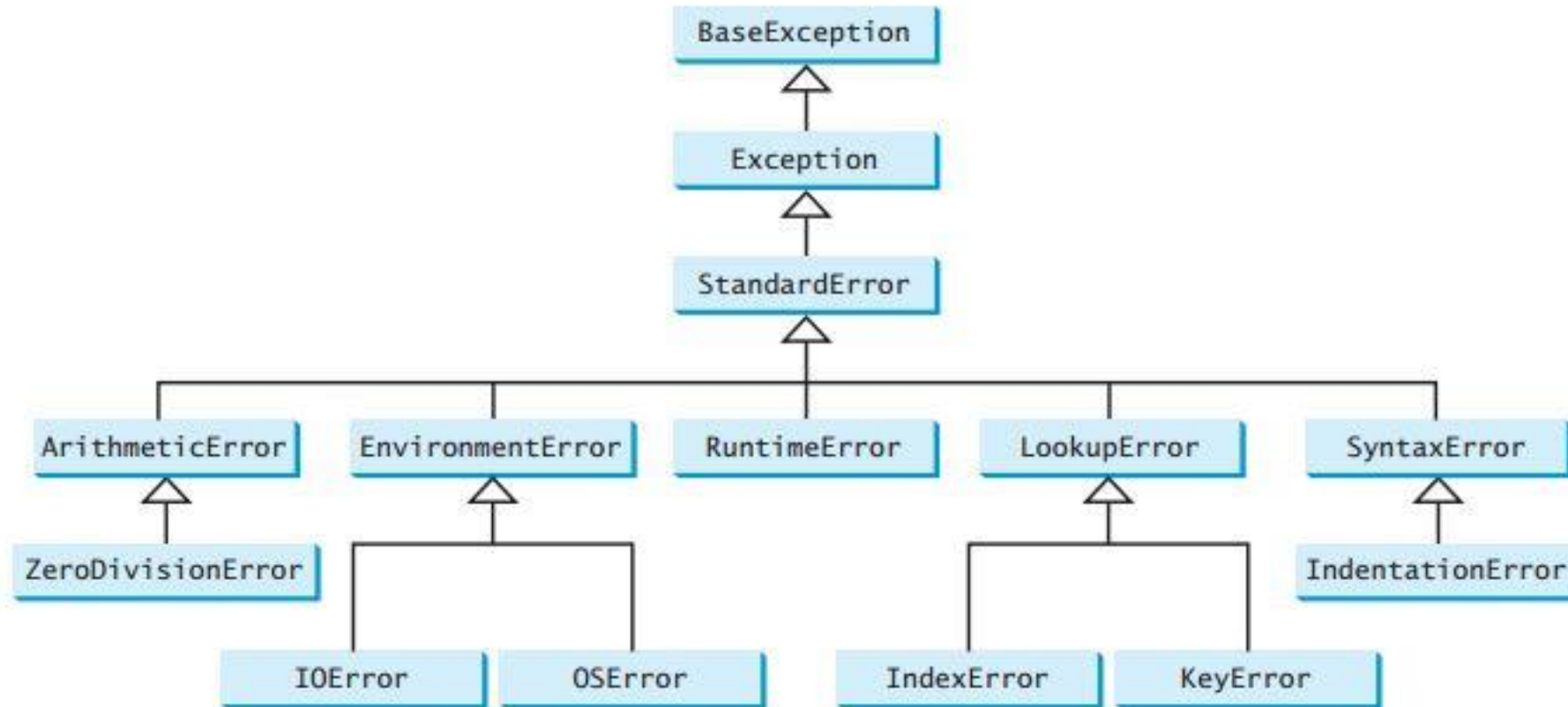
THE CODE DOESN'T WORK...  
**WHY??**



THE CODE IS WORKING...  
**WHY??**



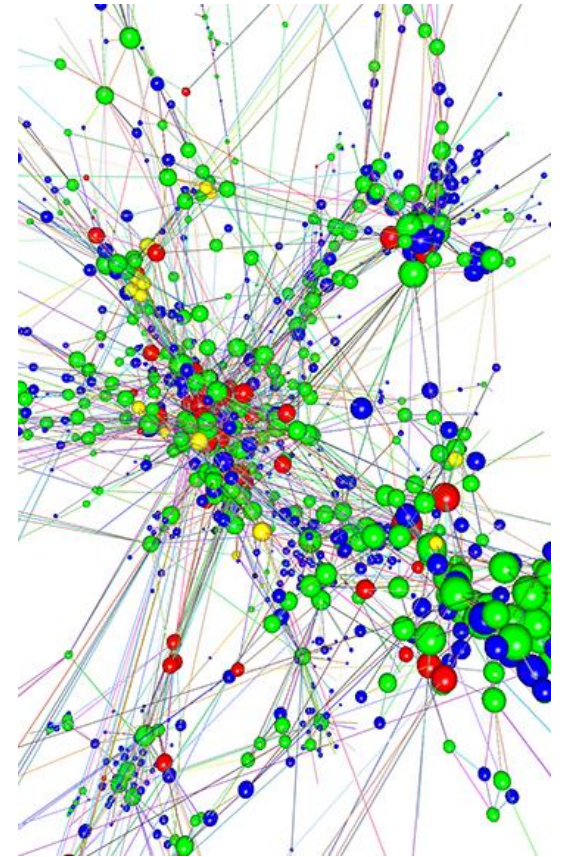
# Excepciones: Jerarquía



# Excepciones: EAFP vs LBYL

¿Qué podemos hacer ante comportamiento inesperado?

- **Look Before You Leap:** Planear que puede salir mal y poner los medios para que no ocurra
- **Easier to Ask Forgiveness than Permission:** En caso de que algo salga mal lo gestionaremos



En Python por lo general **EAFP** (una excepción es un evento raro)

# Excepciones: Ejemplos

```
ex = ZeroDivisionError("Value division by zero")
```

Clase de excepción

Mensaje

`ZeroDivisionError`   `IndexError`   `StopIteration`

