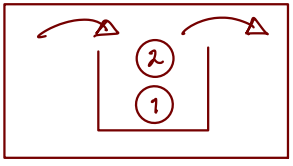
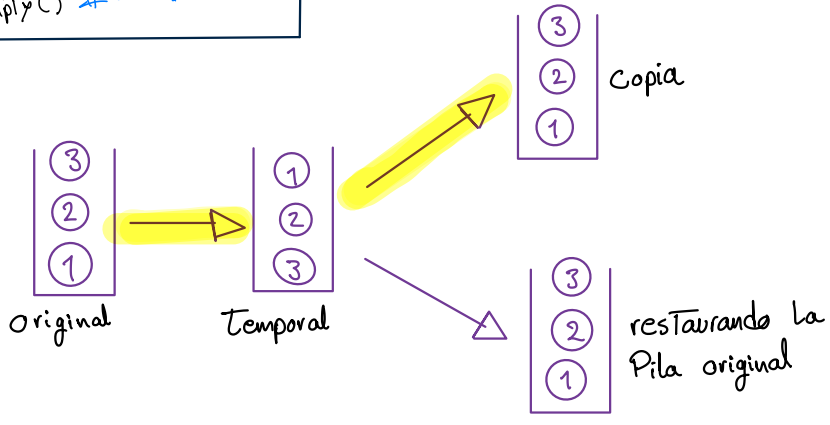
PILAS

from queue import LifoQueue as Pila

```
Variable Cualquiera
↓
p = Pila()
p.put(1) #apilar
elemento = p.get() #desapilar
p.empty() #vacia?
```






```
def copia_pila(pila : Pila) -> Pila :
    temporal : Pila = Pila()
    copia : Pila = Pila()
    while not (pila.empty()) :
        elemento = pila.get()
        temporal.put(elemento)
    while not (temporal.empty()) :
        elem = temporal.get()
        copia.put(elem)
        pila.put(elem)
    return copia
```

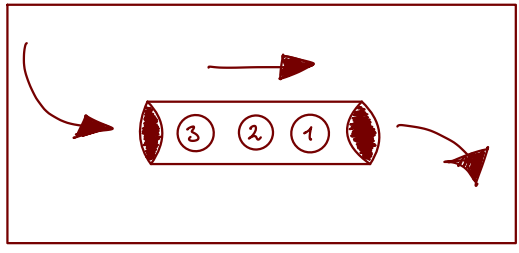
```
pila1 = Pila()
pila1.put(1)
pila1.put(2)
pila1.put(3)
```

[1, 2, 3]

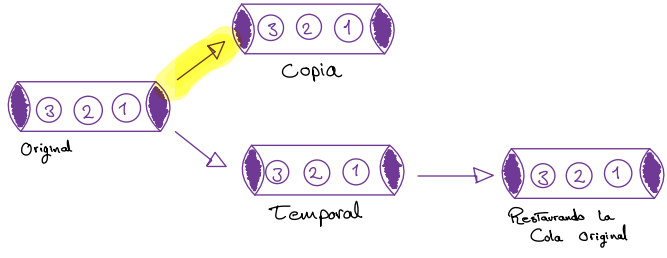
```
Copia = copia_pila(pila1)
print("Copia", copia.queue)
print("original", pila1.queue)
```

COLAS

from queue import Queue as Cola




```
c = Cola()
c.put(1)  #enclar
elemento = c.get()  #desencolar
c.empty()  #vacia?
```




```
def copia_cola(cola:Cola) -> Cola:
    Copia = Cola()
    Temporal = Cola()
    While not(cola.empty()):
        elemento = cola.get()
        Copia.put(elemento)
        Temporal.put(elemento)
    While not(Temporal.empty()):
        elem = Temporal.get()
        cola.put(elem)
    return Copia
```

```
Cola1 = Cola()
Cola1.put(1)
Cola1.put(2)
Cola1.put(3)
```

[1,2,3]

```
Copia = copia_cola(Cola1)
print("Copia", Copia.queue)
print("Original", Cola1.queue)
```

Diccionarios

mi_diccionario = {“nombre”:”jason”, ”edad”:24,
”ocupacion”:”computologo”}

Obtenciones:

1)Obtener todos los pares clave-valor: Los coloca en una lista de Tuplas
-> print(mi_diccionario.items( ))

2)Obtener todos los valores:
-> print(mi_diccionario.values( ))

3)Obtener todas las claves:
-> print(mi_diccionario.keys( ))

5) .pop(clave)
->Elimina un elemento del diccionario y devuelve su valor asociado.

6) .clear( )
->Cuando deseas vaciar completamente un diccionario

Acciones:

1) Acceso a valores:
-> print(mi_diccionario[“nombre”]) -> res = jason

2) Modificación de valores:
-> mi_diccionario[“edad”]) -> res = 20

3) Agregar nuevo par clave-valor:
-> mi_diccionario[“ciudad”]) = “Tokio”

4) Eliminar un par clave-valor:
-> del mi_diccionario[“ocupacion”]

Manejo de Archivos

archivo = open("Path al archivo",Modo,Encoding)

Atributos:

1) "encoding" :

Devuelve la codificación del archivo si se abrió en modo texto.
-> codificacion = archivo.encoding

2) "mode" :

Devuelve el modo en el que el archivo fue abierto.
-> modo = archivo.mode

3) "name" :

Devuelve el nombre del archivo.
-> nombre = archivo.name

3) "closed" :

Devuelve "true" si el archivo está cerrado.
-> estado = archivo.closed

Modos de apertura:

1) "r" (lectura/read) :

Abre el archivo para lectura,el archivo debe existir de lo contrario se genera un error.

2) "w" (escritura/writing):

Abre el archivo para escritura, si el archivo ya existe su contenido se sobreescribe, si el archivo no existe se crea uno nuevo.

3) "a" (agregar/add):

Abre el archivo para agregar contenido al final, si el archivo no existe se crea uno nuevo.

Operaciones básicas:

Lectura de contenido:

1) "read" (size = -1):
Lee y devuelve una cantidad específica de caracteres o bytes del archivo, sino se especifica "size", se lee el contenido completo.

2) "readline" (size = -1):
Lee una línea del archivo. Si se especifica "size", lee hasta "size" caracteres.

3) "readlines" (hint = -1):
Lee todas las líneas del archivo y las devuelve como una lista de cadenas. Si se especifica "hint",intenta leer aproximadamente "hint" bytes.

4) "close( )":
Cierra el archivo.

Escritura de contenido:

1) "write" (texto):
Escribe un texto en el archivo en la posición actual del puntero. Si el archivo ya contiene contenido, se sobrescribe.

2) "writelines" (lineas):
Escribe una lista de líneas(cadenas) en el archivo. Cada línea debe terminar con un salto de línea explícito.