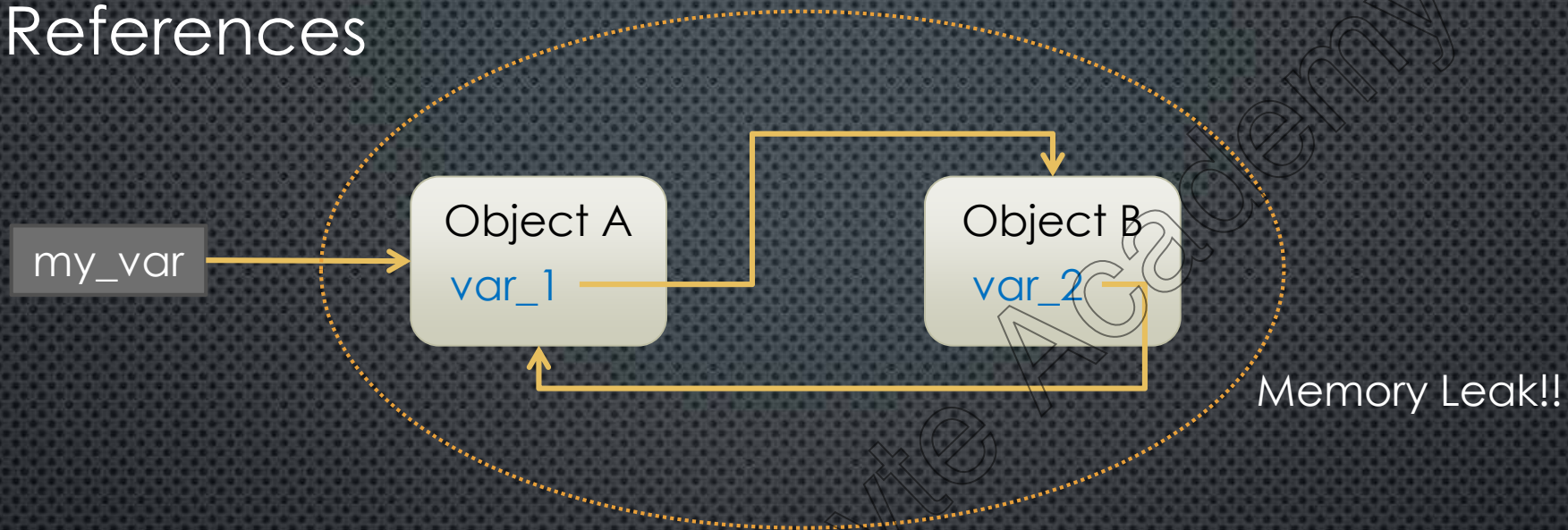


# GARBAGE COLLECTION



# Circular References



## Garbage Collector

- can be controlled programmatically using the **gc** module
- by default it is turned **on**
- you may turn it **off** if you're **sure** your code does not create circular references – but **beware!!**
- runs periodically on its own (if turned on)
- you can call it manually, and even do your own cleanup



In general GC works just fine

but, not always...

## for Python < 3.4

If **even one** of the objects in the circular reference has a destructor [e.g. `__del__()`]

the destruction **order** of the objects may be important

but the GC does not know what that order should be

so the object is marked as **uncollectable**

and the objects in the circular reference are not cleaned up → **memory leak**