

Singleton

September 17, 2024

```
[4]: class mSingleton(type):
    """
    Metaclass for the Singleton class.

    Will hold the singleton and return it rather than a new object
    on instantiation of the class.
    """

    def __call__(cls, *args, **kw):
        """
        Do the singleton magic on instantiation.

        Arguments: cls - the class to instansiate
                  args, kw - various arguments
        Returns: The singleton
        """
        if cls.__instance__ is None:
            cls.__instance__ = type.__call__(cls, *args, **kw)
        return cls.__instance__

class Singleton(metaclass=mSingleton):
    """
    Singleton object mix-in class
    """
    __instance__ = None

class A(Singleton):
    pass

a = A()
b = A()
b is a
```

[4]: True

```
[3]: class X:  
      pass  
x = X()  
y = X()  
x is y
```

[3]: False