super()

super() function returns proxy object
super() function is used to give access to methods and properties of a parent class

Syntax: super()
Parameters: No

```
# super function calls parent constructor
class Parent: # Is subclassed by: Child
   def __init__(self): # Is overridden in: Child
      print("Parent constructor")

class Child(Parent):
   def __init__(self):
      super().__init__()
      print("Child Constructor")

c = Child()
```

```
Parent constructor
Child Constructor
```

```
# super function calls parent constructor
class Parent: # Is subclassed by: Child
    def __init__(self, id, name): # Is overridden in: Child
        print("Parent constructor")
        self.id = id
        self.name = name

    def display(self):
        print(self.id, self.name)

class Child(Parent):
    def __init__(self):
        super().__init__(101, "Admin")
        print("Child Constructor")

c = Child()
c.display()
print(Child.__mro__)
```

```
Parent constructor
Child Constructor
101 Admin
(<class '__main__.Child'>, <class '__main__.Parent'>, <class 'object'>)
```

MRO

Method Resolution Order (MRO) is the order in which methods should be inherited in the presence of multiple inheritance
A method in the derived calls is always called before the method of the base class.

```
# super function calls parent constructor
class GrandParent: # Is subclassed by:   Child
   def __init__(self, id, name): # Is overridden in:   Child
      print("Grand Parent Constructor", id, name)

class Parent(): # Is subclassed by:   Child
   def __init__(self, id, name): # Is overridden in:   Child
      print("Parent Constructor", id, name)

class Child(GrandParent, Parent):
   def __init__(self):
      print("Child Constructor")
      # super().__init__(101, "Admin"), Invokes only grandparent constructor
      GrandParent.__init__(self, 101, "NameOne")
      Parent.__init__(self, 102, "NameTwo")
c = Child()
print(Child.__mro__)
```

```
Child Constructor
Grand Parent Constructor 101 NameOne
Parent Constructor 102 NameTwo
(<class '__main__.Child'>, <class '__main__.GrandParent'>, <class
'__main__.Parent'>, <class 'object'>)
```