

## Super keyword in python

The `super()` keyword in python is used to refer to the parent class. It is especially useful when a class inherits from multiple parent class and you want to call a method from one of the parent classes.

When a class inherits from a parent class, it can override or extend the methods defined in the parent class. However, sometimes you might want to use the parent class method in the child class. This is where `super()` keyword comes in handy.

Here's an example of how to use the `super ()` keyword in a simple inheritance scenario.

```
class ParentClass:
    def parent_method(self):
        print("This is the parent method")

class childClass(ParentClass):

    def parent_method(self):
        print("Munawar")
        super().parent_method()

    def child_method(self):
        print("This is the child method")
        super().parent_method()

obj=childClass()
obj.child_method()
```

Here's an example of how to use the `super ()` keyword with the constructor:

```
# use super with constructor
class Programmer:
    def __init__(self,name,id):
        self.name=name
        self.id=id

class framework:
```

```

def __init__(self,language,code,framework):
    # super().__init__(name,id)
    self.language=language
    self.code=code
    self.framework=framework

munawar=Programmer("Munawar Johar",148)
kamal=framework("Javascript","junior coder","Angular")

print(kamal.language)
print(kamal.code)
print(kamal.framework)

```

## Source Code

```

# class ParentClass:
#     def parent_method(self):
#         print("This is the parent method")

# class childClass(ParentClass):

#     def parent_method(self):
#         print("Munawar")
#         super.parent_method()

#     def child_method(self):
#         print("This is the child method")
#         super().parent_method()

# obj=childClass()
# obj.child_method()

# use super with constructor
class Programmer:
    def __init__(self,name,id):
        self.name=name
        self.id=id

```

```
class framework:
    def __init__(self,language,code,framework):
        # super().__init__(name,id)
        self.language=language
        self.code=code
        self.framework=framework

munawar=Programmer("Munawar Johar",148)
kamal=framework("Javascript","junior coder","Angular")

print(kamal.language)
print(kamal.code)
print(kamal.framework)
```

Thank You