Multiple Inheritance in Python

Multiple inheritance is a feature in object-oriented programming where a class can inherit attributes and methods from more than one parent class. Python supports multiple inheritance, allowing for flexible and reusable code structures.

In Python, a class can inherit from multiple base classes by listing them in parentheses.

Example:

```
class ParentA:
    def method_a(self):
        print("Method from ParentA")

class ParentB:
    def method_b(self):
        print("Method from ParentB")

class Child(ParentA, ParentB):
    pass

child_instance = Child()
child_instance.method_a()  # Output: Method from ParentA
child_instance.method_b()  # Output: Method from ParentB
```

Advantages of Multiple Inheritance

- Code Reusability: Reuse methods and attributes from multiple classes.
- Modularity: Separate concerns into different classes and combine them.
- Flexibility: Combine functionalities from different sources.

Conclusion

Multiple inheritance in Python is a powerful tool that allows a class to inherit from multiple sources. While it provides flexibility and promotes code reuse, it also introduces complexity such as the diamond problem. Understanding Python's MRO helps in resolving such ambiguities effectively.