In Python, there are three main types of inheritance: single inheritance, multiple inheritance, and multilevel inheritance. Each type allows classes to inherit attributes and methods from other classes in different ways. Here's an explanation of each type with examples:

* Single inheritance involves a child class inheriting from only one parent class.
* Multiple inheritance allows a child class to inherit from multiple parent classes.
* Multilevel inheritance involves a chain of inheritance where a child class inherits from a parent class, and another class inherits from the child class.

**Single Inheritance:**

class Parent:

def parent\_method(self):

print("Parent method")

class Child(Parent):

def child\_method(self):

print("Child method")

# Create an instance of Child

child\_instance = Child()

# Child instance can access methods from both Child and Parent classes

child\_instance.child\_method()

child\_instance.parent\_method()

**Multiple Inheritance:**

class Parent1:

def method1(self):

print("Method from Parent1")

class Parent2:

def method2(self):

print("Method from Parent2")

class Child(Parent1, Parent2):

def child\_method(self):

print("Child method")

# Create an instance of Child

child\_instance = Child()

# Child instance can access methods from Child, Parent1, and Parent2 classes

child\_instance.child\_method()

child\_instance.method1()

child\_instance.method2()

**Multilevel Inheritance**:

class Grandparent:

def grandparent\_method(self):

print("Grandparent method")

class Parent(Grandparent):

def parent\_method(self):

print("Parent method")

class Child(Parent):

def child\_method(self):

print("Child method")

# Create an instance of Child

child\_instance = Child()

# Child instance can access methods from Child, Parent, and Grandparent classes

child\_instance.child\_method()

child\_instance.parent\_method()

child\_instance.grandparent\_method()