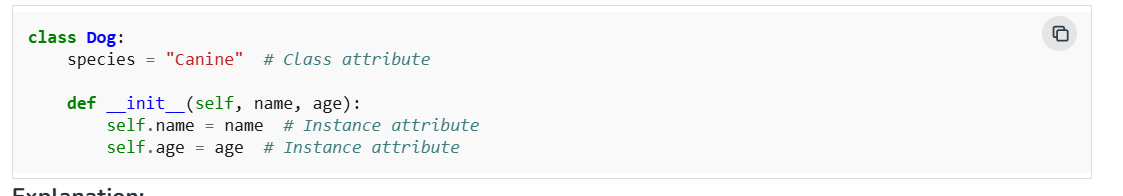
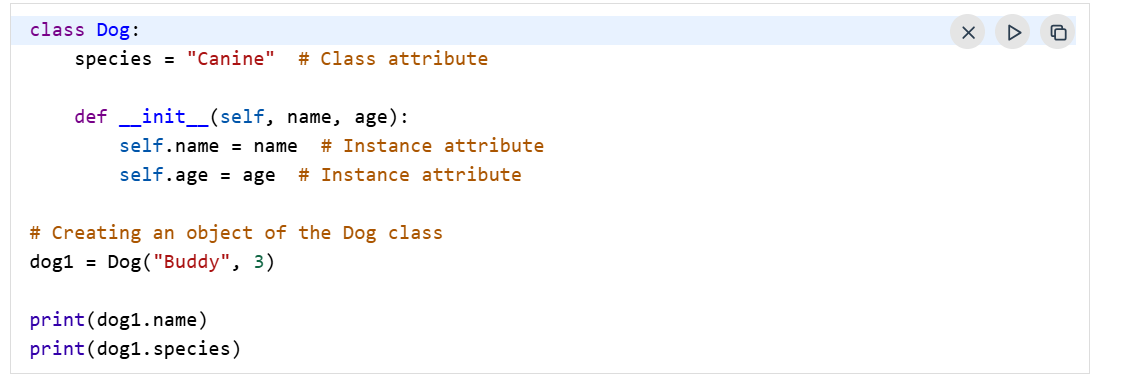
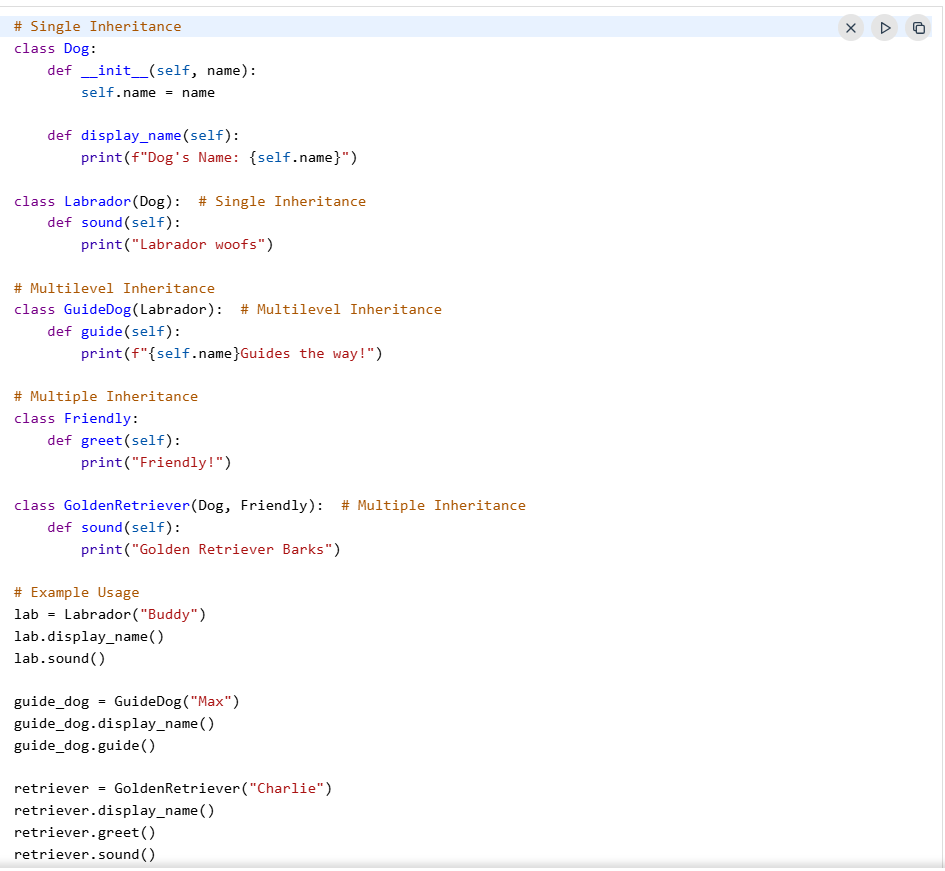
## **Python Class**

Classes are blueprints for creating objects  
  


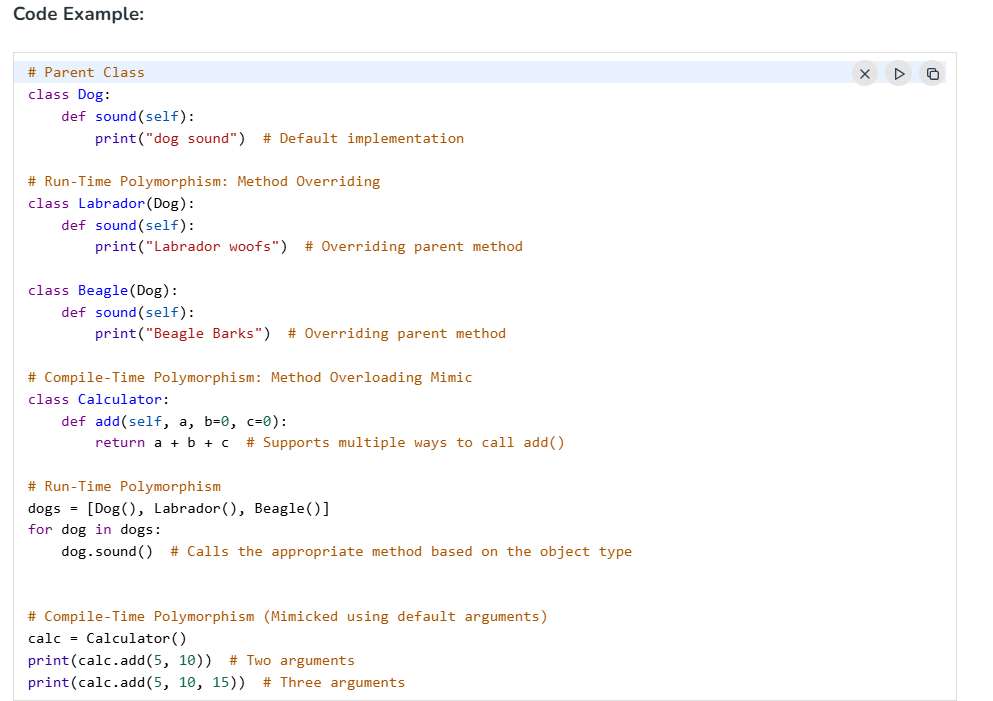
## **Python Objects**

An Object is an instance of a Class; it represents a specific implementation of the class and holds its own data.   
  
  
**Python Inheritance**

One class can reuse code from another (parent → child).  
  
  
**Python Polymorphism**

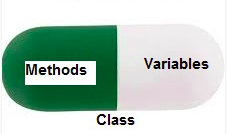
1. Polymorphism allows methods to have the same name but behave differently based on the object's context.  
     
    **Compile-Time Polymorphism**:

**Run-Time Polymorphism**:

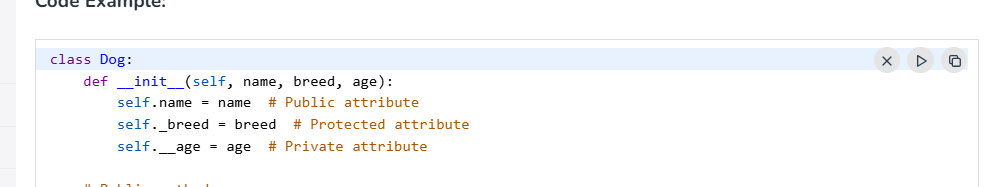
  
  
**Python Encapsulation**

Encapsulation will Wrap the attributes and methods within a class, restricting access to some components to control interactions.

A class is an example of encapsulation as it encapsulates all the data that is member functions, variables, etc.



### **Types of Encapsulations:**

1. **Public Members**: Accessible from anywhere.
2. **Protected Members**: Accessible within the class and its subclasses.
3. **Private Members**: Accessible only within the class.  
   

## **Data Abstraction**

Abstraction hides the internal implementation details while exposing only the necessary functionality. It helps focus on "what to do" rather than "how to do it."  
  


\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
**🔹 1. Instance Method (default)**

* **Takes self as the first argument**.
* Can access or modify the object’s state (instance variables).

python

CopyEdit

class MyClass:  
 def instance\_method(self):  
 print("This is an instance method")  
  
  
**🔹 2. Class Method**

* Defined using @classmethod decorator.
* Takes **cls** (class) as the first argument instead of self.
* Can access or modify the class state (class variables).

python

CopyEdit

class MyClass:  
 name = "Python"  
  
 @classmethod  
 def show\_name(cls):  
 print("Class name is:", cls.name)  
  
MyClass.show\_name() # ✅ works without creating an object

**🔹 3. Static Method**

* Defined using @staticmethod decorator.
* **Does not take self or cls** as the first argument.
* Cannot access class or instance variables directly.
* Used for utility/helper functions.

python

CopyEdit

class MyClass:  
 @staticmethod  
 def add(a, b):  
 return a + b  
  
print(MyClass.add(3, 5)) # ✅ 8  
  
  
 ------------  
  
