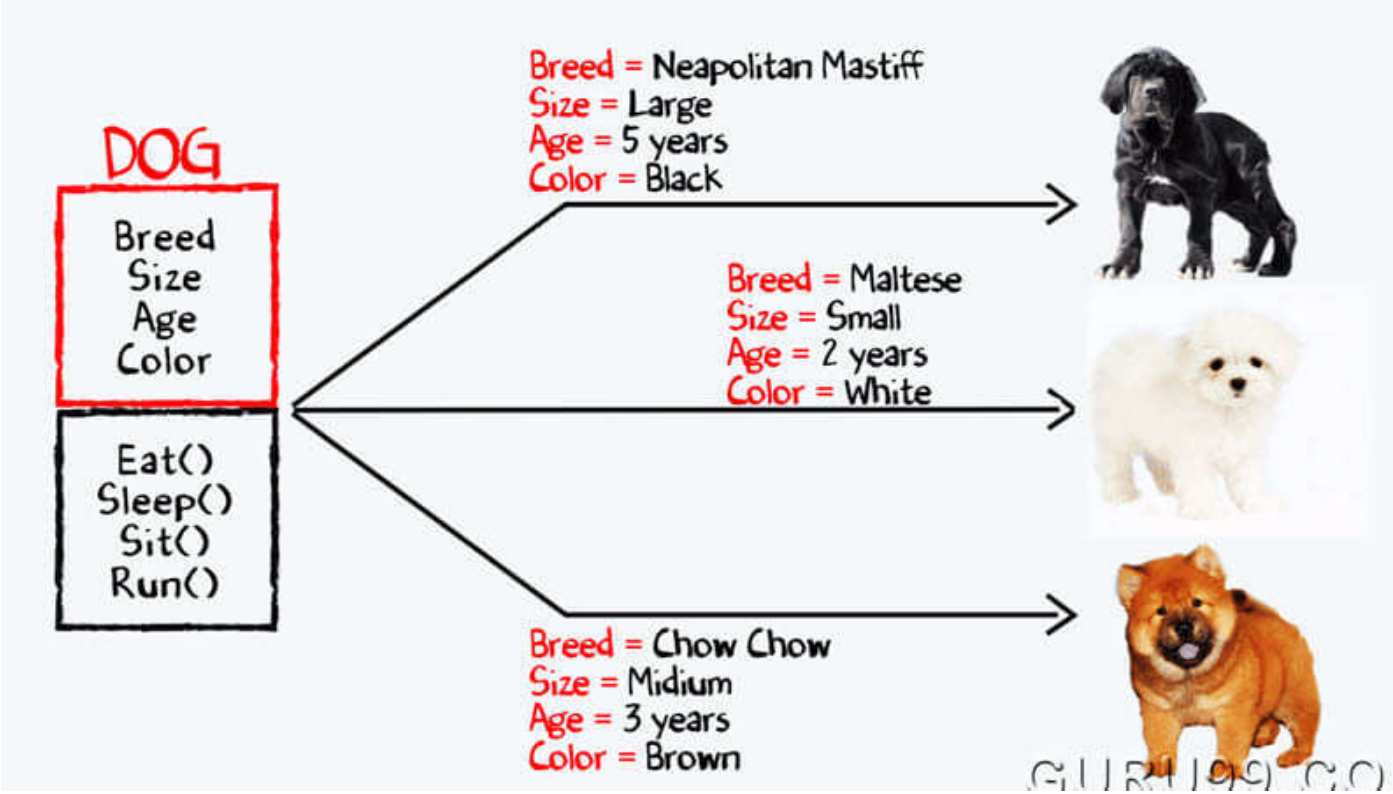
Class and Object

- **Class**
  - A **class** is a **blueprint** or **template** for creating objects.
  - It describes **what data** (attributes/data fields) and **what actions** (methods) the objects will have.
  - A class itself is **not** an actual thing — it's just a design.
- **Object**
  - An **object** is an **actual instance** created from a class.
  - It's a **real thing in memory** with its own data.
  - You can create many objects from the same class, each with different values.

Class and Object



Key Points for Defining a Class in Python

1. Use the class keyword
   - Start with class followed by the class name (first letter usually capitalized)

```
class Student:
```

2. Attributes (Data Fields)
   - Variables that store data inside the object.
   - Usually defined and initialized in the constructor (__init__() method).

```
    def __init__(self, name, age):
        self.name = name
        self.age = age
```

3. Methods (Functions inside the class)
   - Actions the object can perform.
   - Always include self as the first parameter to work with that object's data.

```
    def greet(self):
        print("Hello, my name is", self.name)
```

Self

- In Python classes, self is a reference to the current object.
- It lets the method know which object's data it should work with.
- Python automatically passes self as the first argument to instance methods.

```python
class Student:
    def __init__(self, name):
        self.name = name  # store the name in this object's data field

    def greet(self):
        print("Hello, my name is", self.name)

# Create two student objects
s1 = Student("Alice")
s2 = Student("Bob")

s1.greet()  # self → s1   → "Hello, my name is Alice"
s2.greet()  # self → s2   → "Hello, my name is Bob"
```

An object has its own attributes and actions.
In a class, since we don't have a specific object yet, we use `self` to represent the object.

Constructor

- __init__ Method
  - Special built-in method that runs automatically when an object is created.
  - Used to **initialize attributes**.
  - When creating an object, the arguments you pass must match the parameters in the __init__() method — excluding self.

```python
class Student:
    def __init__(self, name, age):  # 2 parameters (excluding self)
        self.name = name
        self.age = age

# When creating object → pass 2 arguments, NOT including self
s1 = Student("Alice", 17)  # ✅ matches name, age
```