Instance and Class variables in python

In Python, classes can have two types of variables: instance variables and class variables. Understanding the difference between these two is crucial for managing the state and behavior of your objects properly.

Instance Variables

Instance variables are attributes that are specific to each instance of a class. They are defined within methods (typically the `__init__` method) and are prefixed with `self` to denote that they belong to the instance.

Defining Instance Variables

Instance variables are usually initialized inside the `__init__` method, which is the constructor for the class. Each instance of the class will have its own copy of these variables.

Example:

```python
class Dog:
    def __init__(self, name, age):
        self.name = name  # Instance variable
        self.age = age    # Instance variable

# Creating two instances of Dog
dog1 = Dog("Buddy", 3)
dog2 = Dog("Molly", 5)

# Accessing instance variables
print(dog1.name)  # Output: Buddy
print(dog2.name)  # Output: Molly
```

In this example, `name` and `age` are instance variables. `dog1` and `dog2` have their own separate `name` and `age` attributes.

Class Variables

Class variables are attributes that are shared by all instances of a class. They are defined within the class but outside any methods. All instances of the class share the same value for class variables.

Defining Class Variables

Class variables are typically defined directly beneath the class definition and outside of any instance methods.

Example:

```
class Dog:
    # Class variable
    species = "Canis familiaris"

    def __init__(self, name, age):
        self.name = name  # Instance variable
        self.age = age    # Instance variable

# Creating instances of Dog
dog1 = Dog("Buddy", 3)
dog2 = Dog("Molly", 5)

# Accessing class variables
print(dog1.species)  # Output: Canis familiaris
print(dog2.species)  # Output: Canis familiaris

# Modifying the class variable
Dog.species = "Canis lupus"

print(dog1.species)  # Output: Canis lupus
print(dog2.species)  # Output: Canis lupus
```

In this example, `species` is a class variable. Changing `Dog.species` affects all instances of the class.

Differences Between Instance and Class Variables

1. **Scope**:
   - **Instance Variables**: Scoped to the instance; each instance has its own copy.
   - **Class Variables**: Scoped to the class; shared across all instances.
2. **Initialization**:
   - **Instance Variables**: Initialized inside methods (usually `__init__`) with `self`.
   - **Class Variables**: Initialized directly within the class definition.
3. **Access**:
   - **Instance Variables**: Accessed and modified through `self`.
   - **Class Variables**: Accessed through the class name or instances, but typically modified through the class name.

Example: Combining Instance and Class Variables

CODE:

```
class Dog:
    # Class variable
    species = "Canis familiaris"

    def __init__(self, name, age):
        self.name = name  # Instance variable
        self.age = age    # Instance variable

    def details(self):
        return f"{self.name} is {self.age} years old and belongs to the
species {Dog.species}"

# Creating instances of Dog
dog1 = Dog("Buddy", 3)
dog2 = Dog("Molly", 5)

# Accessing instance and class variables
print(dog1.details())  # Output: Buddy is 3 years old and belongs to the
species Canis familiaris
print(dog2.details())  # Output: Molly is 5 years old and belongs to the
species Canis familiaris

# Modifying class variable
Dog.species = "Canis lupus"

print(dog1.details())  # Output: Buddy is 3 years old and belongs to the
species Canis lupus
print(dog2.details())  # Output: Molly is 5 years old and belongs to the
species Canis lupus
```

Output:

```
C:\Users\attacker\PycharmProjects\pythonProject\.venv\Scripts\python.exe C:\Users\attacker\PycharmProjects\pythonProject\30.py
Buddy is 3 years old and belongs to the species Canis familiaris
Molly is 5 years old and belongs to the species Canis familiaris
Buddy is 3 years old and belongs to the species Canis lupus
Molly is 5 years old and belongs to the species Canis lupus
```

Summary

- **Instance Variables**: Specific to each instance, defined within methods using `self`.
- **Class Variables**: Shared among all instances, defined directly within the class.
- Use instance variables for data unique to each instance.
- Use class variables for data shared across all instances.

Understanding these concepts helps in designing classes that maintain proper state and behavior, providing a clear structure to your code.