



Instance variables vs Class variables in Python | Python Tutorial - Day #66



[Overview](#) [Q&A](#) [Downloads](#) [Announcements](#)

Day 66 – Instance vs class variables

In Python, variables can be defined at the class level or at the instance level. Understanding the difference between these types of variables is crucial for writing efficient and maintainable code.

Class Variables

Class variables are defined at the class level and are shared among all instances of the class. They are defined outside of any method and are usually used to store information that is common to all instances of the class. For example, a class variable can be used to store the number of instances of a class that have been created.

```
class MyClass:
    class_variable = 0

    def __init__(self):
        MyClass.class_variable += 1

    def print_class_variable(self):
        print(MyClass.class_variable)

obj1 = MyClass()
obj2 = MyClass()

obj1.print_class_variable() # Output: 2
```

In the example above, the `class_variable` is shared among all instances of the class `MyClass`. When we create new instances of `MyClass`, the value of `class_variable` is incremented. When we call the `print_class_variable` method on `obj1` and `obj2`, we get the same value of `class_variable`.

Instance Variables

Instance variables are defined at the instance level and are unique to each instance of the class. They are defined inside the **`init`** method and are usually used to store information that is specific to each instance of the class. For example, an instance variable can be used to store the name of an employee in a class that represents an employee.

```
class MyClass:
    def __init__(self, name):
```

```
        self.name = name

def print_name(self):
    print(self.name)

obj1 = MyClass("John")
obj2 = MyClass("Jane")

obj1.print_name() # Output: John
obj2.print_name() # Output: Jane
```

In the example above, each instance of the class `MyClass` has its own value for the `name` variable. When we call the `print_name` method on `obj1` and `obj2`, we get different values for `name`.

Summary

In summary, class variables are shared among all instances of a class and are used to store information that is common to all instances. Instance variables are unique to each instance of a class and are used to store information that is specific to each instance. Understanding the difference between class variables and instance variables is crucial for writing efficient and maintainable code in Python.

It's also worth noting that, in python, class variables are defined outside of any methods and don't need to be explicitly declared as class variable. They are defined in the class level and can be accessed via `classname.variable_name` or `self.class.variable_name`. But instance variables are defined inside the methods and need to be explicitly declared as instance variable by using `self.variable_name`.

[Previous](#)

[Next](#)



CodeWithHarry

Copyright © 2023 CodeWithHarry.com

