# Who Can Access What? – Exploring Access Modifiers in Python

## Objective:

To explore how access modifiers (public, protected, private) work in Python classes using attributes and observe their effect on object access and data visibility.

## Instructions:

1. Review the code provided below.
2. Predict the output of each print() statement.
3. Run the code in your Python environment.
4. Answer the reflection questions based on what you observed.

## 🧪 Code Snippet:

class Person:  
 def \_\_init\_\_(self, name, age, ssn):  
 self.name = name # Public attribute  
 self.\_age = age # Protected attribute  
 self.\_\_ssn = ssn # Private attribute  
  
 def show\_info(self):  
 print(f"Name: {self.name}")  
 print(f"Age: {self.\_age}")  
 print(f"SSN: {self.\_\_ssn}") # Accessing private from inside the class  
  
# Object creation  
person1 = Person("Dan", 30, "123-45-6789")  
  
# External access  
print("Public:", person1.name)  
print("Protected:", person1.\_age)  
# print("Private:", person1.\_\_ssn) # Uncomment to test  
# print("Private (mangled):", person1.\_Person\_\_ssn) # Uncomment to test

## Step 1: Predictions

|  |  |
| --- | --- |
| Code Line | What do you think will happen? (Write your prediction) |
| print("Public:", person1.name) | It will print “Dan” |
| print("Protected:", person1.\_age) | It will print 30 but with warning |
| print("Private:", person1.\_\_ssn) | It will cause an error because ssn is a private attribute |
| print("Private (mangled):", person1.\_Person\_\_ssn) | It produced an Error |

## Step 2: Run and Observe

|  |  |
| --- | --- |
| Code Line | Actual Result / Error Message |
| print("Public:", person1.name) | Public : Dan |
| print("Protected:", person1.\_age) | Protected : 30 |
| print("Private:", person1.\_\_ssn) | AttributeError: 'Person' object has no attribute '\_\_ssn' |
| print("Private (mangled):", person1.\_Person\_\_ssn) | Private (mangled): 123-45-6789 |

## Step 3: Reflection Questions

1. What is the difference between public, protected, and private attributes?

- Public:

- Protected:

- Private:

2. Why was accessing \_\_ssn directly outside the class an error?

3. How was the \_\_ssn attribute still accessible from within the class?

4. What do we mean by name mangling in Python?

5. Why might we use private or protected attributes instead of making everything public?

## Reminder: Python Access Modifier Conventions

|  |  |  |
| --- | --- | --- |
| Modifier | Syntax | Access Level Description |
| Public | self.name | Can be accessed anywhere |
| Protected | self.\_age | Meant for internal use/subclasses (still accessible) |
| Private | self.\_\_ssn | Not directly accessible outside the class (name-mangled) |