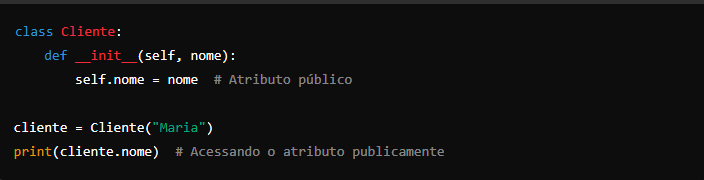
O encapsulamento é um dos princípios fundamentais da Programação Orientada a Objetos (POO) e consiste em restringir o acesso direto a certos atributos ou métodos de uma classe, permitindo o controle sobre como esses dados são acessados ou modificados. Em Python, o encapsulamento é implementado usando modificadores de acesso.

Aqui estão os principais conceitos do encapsulamento em Python:

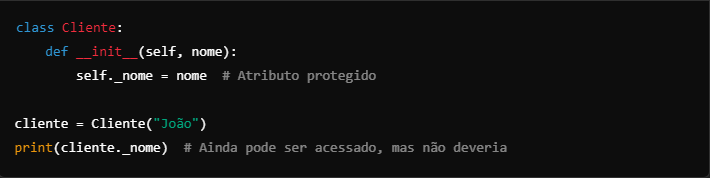
**1. Atributos Públicos**

Esses atributos são acessíveis de qualquer lugar, dentro ou fora da classe. Por padrão, todos os atributos em Python são públicos.



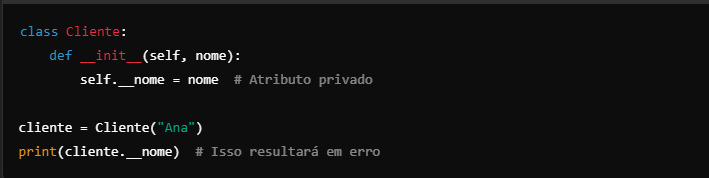
### 2. ****Atributos Protegidos****

Em Python, um atributo é considerado protegido se seu nome começa com um único sublinhado (\_). Isso é apenas uma convenção, não uma regra rígida; esses atributos ainda podem ser acessados de fora da classe, mas indica que não deveriam ser modificados diretamente.



### 3. ****Atributos Privados****

Para tornar um atributo privado, seu nome deve começar com dois sublinhados (\_\_). Isso impede o acesso direto ao atributo fora da classe. Python utiliza "name mangling" (ofuscação do nome) para evitar que o atributo seja facilmente acessado.

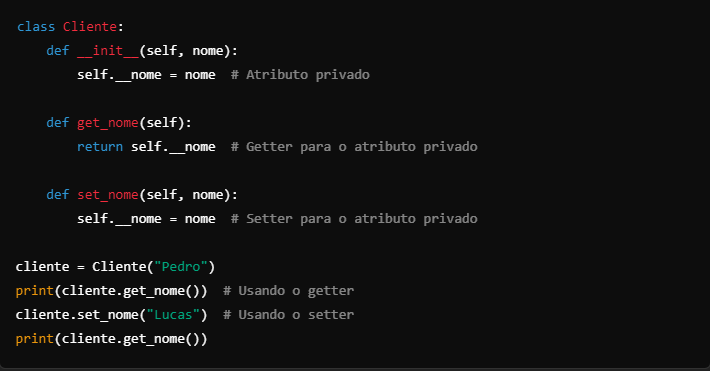


No entanto, é possível acessar atributos privados indiretamente utilizando métodos especiais ou através de um truque, usando o nome "mangleado" (\_Classe\_\_atributo).



### 4. ****Uso de Getter e Setter****

Para acessar ou modificar atributos privados, você normalmente define métodos "getter" e "setter", que são responsáveis por retornar e modificar o valor do atributo, mantendo o controle de como os dados são manipulados.



### Benefícios do Encapsulamento

* **Proteção dos Dados**: Permite controlar o acesso e modificação de atributos, garantindo que os dados estejam sempre em um estado válido.
* **Flexibilidade**: Os métodos getters e setters oferecem mais flexibilidade para alterar a implementação interna sem impactar o código que utiliza a classe.
* **Manutenção**: Facilita a manutenção e entendimento do código, limitando o acesso direto a dados sensíveis.

Esse conceito de encapsulamento será útil no seu projeto de cadastro de clientes, já que permitirá controlar e proteger os dados que cada instância de cliente possui.