



## ○ O que são atributos de visibilidade?

Os atributos de visibilidade definem o nível de acesso de uma característica ou comportamento dentro de uma classe. Como o próprio nome já diz, este recurso visa garantir o nível de visibilidade do código definido dentro de uma classe.

Existem três tipos de atributos de visibilidade nas linguagens orientadas a objetos, que são:

- **Public:** Atributos e métodos definidos como públicos poderão ser invocados, acessados e modificados através de qualquer lugar do projeto;
- **Private:** Atributos e métodos definidos como privados só poderão ser invocados, acessados e modificados somente por seu próprio objeto.
- **Protected:** Atributos e métodos definidos como protegidos só poderão ser invocados, acessados e modificados por classes que herdaram de outras classes através do conceito de Herança, visto na última aula. Sendo assim, apenas classes "filhas" poderão acessar métodos e atributos protegidos.

No Python, diferente das linguagens completamente voltadas ao paradigma da orientação à objetos (Java, C#, etc.), estes atributos existem, mas não da forma "convencional".

No Java, por exemplo, ao declarar um atributo privado este só será acessível por seu próprio objeto. Já no Python, este conceito é um pouco diferente. Por convenção, quando queremos definir que um atributo deve ser privado, adicionamos dois underlines (\_\_) na frente do nome da variável, fazendo com que quem vá mexer em seu código entenda, se ele conhece a convenção, que o atributo só deve ser acessado por seu próprio objeto.

E este conceito vale para todos os outros tipos de atributos de visibilidade. Para o atributo protected, adicionamos um underline (\_) para indicar que um atributo só poderá ser acessado por suas classes "filhas".

