

# Python - Classes Internas

Uma classe definida dentro de outra classe é conhecida como classe interna em Python. Às vezes, a classe interna também é chamada de classe aninhada. Se a classe interna for instanciada, o objeto da classe interna também poderá ser usado pela classe pai. O objeto da classe interna torna-se um dos atributos da classe externa. A classe interna herda automaticamente os atributos da classe externa sem estabelecer formalmente a herança.

## Sintaxe

```
class outer:
    def __init__(self):
        pass
    class inner:
        def __init__(self):
            pass
```

Uma classe interna permite agrupar classes. Uma das vantagens de aninhar classes é que fica fácil entender quais classes estão relacionadas. A classe interna tem um escopo local. Ele atua como um dos atributos da classe externa.

## Exemplo

No código a seguir, temos aluno como classe externa e assuntos como classe interna. O construtor `__init__()` do aluno inicializa o atributo `name` e uma instância da classe `subject`. Por outro lado, o construtor da classe de assuntos internos inicializa duas variáveis de instância `sub1`, `sub2`.

Um método `show()` da classe externa chama o método da classe interna com o objeto que foi instanciado.

```
class student:
    def __init__(self):
        self.name = "Ashish"
        self.subs = self.subjects()
        return
    def show(self):
        print("Name:", self.name)
        self.subs.display()
    class subjects:
        def __init__(self):
            self.sub1 = "Phy"
```



```
        self.sub2 = "Che"
    return
def display(self):
    print ("Subjects:",self.sub1, self.sub2)

s1 = student()
s1.show()
```

Quando você executa este código, ele produzirá a seguinte **saída** -

```
Name: Ashish
Subjects: Phy Che
```

É bem possível declarar um objeto de classe externa de forma independente e fazê-lo chamar seu próprio método display().

```
sub = student().subjects().display()
```

Ele listará os assuntos.