

Python - Atributos de Classe

Cada classe Python mantém os seguintes atributos integrados e eles podem ser acessados usando o operador ponto como qualquer outro atributo -

- **__dict__** - Dicionário contendo o namespace da classe.
- **__doc__** - String de documentação da classe ou nenhuma, se indefinida.
- **__name__** - Nome da classe.
- **__module__** - Nome do módulo no qual a classe está definida. Este atributo é "__main__" no modo interativo.
- **__bases__** - Uma tupla possivelmente vazia contendo as classes base, na ordem de ocorrência na lista de classes base.

Para a classe acima, vamos tentar acessar todos esses atributos -

```
class Employee:
    def __init__(self, name="Bhavana", age=24):
        self.name = name
        self.age = age
    def displayEmployee(self):
        print ("Name : ", self.name, ", age: ", self.age)

print ("Employee.__doc__:", Employee.__doc__)
print ("Employee.__name__:", Employee.__name__)
print ("Employee.__module__:", Employee.__module__)
print ("Employee.__bases__:", Employee.__bases__)
print ("Employee.__dict__:", Employee.__dict__ )
```

Ele produzirá a seguinte **saída** -

```
Employee.__doc__: None
Employee.__name__: Employee
Employee.__module__: __main__
Employee.__bases__: (<class 'object'>,)
Employee.__dict__: {'__module__': '__main__', '__init__': <function Employee.__init__ at 0x0000000000000000>, 'displayEmployee': <function Employee.displayEmployee at 0x0000000000000000>}
```

Variáveis de classe



No exemplo da classe `Employee` acima, nome e idade são variáveis de instância, pois seus valores podem ser diferentes para cada objeto. Um atributo ou variável de classe cujo valor é compartilhado entre todas as instâncias de a nesta classe. Um atributo de classe representa um atributo comum de todos os objetos de uma classe.

Os atributos de classe não são inicializados dentro do construtor `__init__()`. Eles são definidos na classe, mas fora de qualquer método. Eles podem ser acessados pelo nome da classe além do objeto. Em outras palavras, um atributo de classe disponível para a classe, bem como para o seu objeto.

Exemplo

Vamos adicionar uma variável de classe chamada **`empCount`** na classe `Employee`. Para cada objeto declarado, o método `__init__()` é chamado automaticamente. Este método inicializa as variáveis de instância e também incrementa **`empCount`** em 1.

```
class Employee:
    empCount = 0
    def __init__(self, name, age):
        self.__name = name
        self.__age = age
        Employee.empCount += 1
        print ("Name: ", self.__name, "Age: ", self.__age)
        print ("Employee Number:", Employee.empCount)

e1 = Employee("Bhavana", 24)
e2 = Employee("Rajesh", 26)
e3 = Employee("John", 27)
```

Saída

Declaramos três objetos. Cada vez, o **`empCount`** aumenta em 1.

```
Name: Bhavana Age: 24
Employee Number: 1
Name: Rajesh Age: 26
Employee Number: 2
Name: John Age: 27
Employee Number: 3
```