Python - Classe e Objetos Anônimos

A função type() integrada do Python retorna a classe à qual um objeto pertence. Em Python, uma classe, tanto uma classe interna quanto uma classe definida pelo usuário, são objetos do tipo class.

Exemplo

```
class myclass:
   def __init__(self):
      self.myvar=10
      return

obj = myclass()

print ('class of int', type(int))
print ('class of list', type(list))
print ('class of dict', type(dict))
print ('class of myclass', type(myclass))
print ('class of obj', type(obj))
```

Ele produzirá a seguinte **saída** -

```
class of int <class 'type'>
class of list <class 'type'>
class of dict <class 'type'>
class of myclass <class 'type'>
```

O type() tem uma versão de três argumentos como segue -

Sintaxe

```
newclass=type(name, bases, dict)
```

Usando a sintaxe acima, uma classe pode ser criada dinamicamente. Três argumentos do tipo função são -

- **name** - nome da classe que se torna o atributo __name__ da nova classe
- **bases** - tupla consistindo em classes pai. Pode ficar em branco se não for uma c derivada

- **dict** - dicionário formando namespace da nova classe contendo atributos e métodos e seus valores.

Podemos criar uma classe anônima com a versão acima da função type(). O argumento do nome é uma string nula, o segundo argumento é uma tupla de uma classe da classe do objeto (observe que cada classe em Python é herdada da classe do objeto). Adicionamos certas variáveis de instância como o dicionário do terceiro argumento. Nós o mantemos vazio por enquanto.

```
anon=type('', (object, ), {})
```

Para criar um objeto desta classe anônima -

```
obj = anon()
print ("type of obj:", type(obj))
```

O resultado mostra que o objeto é de classe anônima

```
type of obj: <class '__main__.'>
```

Exemplo

Também podemos adicionar variáveis de instância e métodos de instância dinamicamente. Dê uma olhada neste exemplo -

```
def getA(self):
   return self.a
obj = type('',(object,),{'a':5,'b':6,'c':7,'getA':getA,'getB':lambda self : self.b})(
print (obj.getA(), obj.getB())
```

Ele produzirá a seguinte **saída** -

```
5 6
```