

10

Descobrindo o tipo de um objeto

Você se lembra da função `type` para descobrir o tipo de um objeto? Não? Então vamos revisar!

No capítulo de tuples e listas testamos a função `type`:

```
>>> nomes = ['Flavio', 'Nico', 'Fabio']
>>> type(nomes)
<type 'list'>
```

Agora já podemos usar o nosso novo vocabulário e dizer que lista também é um objeto! Só que nesse caso não é preciso chamar uma função construtora. Há uma sintaxe especial (`[]`) para criar esse objeto. A função `type`, por sua vez, descobre qual é o tipo desse objeto.

Será que isso também funciona para objetos da classe `Perfil`? Ou seja, você já tem dentro do arquivo `models.py` a classe `Perfil`:

```
class Perfil():
    'Classe padrão para perfis de usuários'
    def __init__(self, nome, telefone, empresa):
        self.nome = nome
        self.telefone = telefone
        self.empresa = empresa
```

E que no console do Python tenhamos feito:

```
>>> from models import Perfil
>>> perfil = Perfil('Steve', '2222-3333', 'Minecraft')
```

Agora tente executar:

```
>>> type(perfil)
```

Qual é a saída?

Seleciona uma alternativa

A Perfil

B Não funciona.

C <type 'instance'>

D

```
<type 'models.Perfil'>
```

E

Algo como `Perfil@2131231`