

06

Deixando mais orientado à objetos

Transcrição

Nesse ponto, queremos implementar mais informações na nossa página, por exemplo, a categoria do jogo (como "Luta" ou "Esporte") e qual o console dele.

Como fazemos para ter, no Python, várias características de um mesmo conceito — `jogo`, nesse caso? Precisaremos criar uma classe, ou seja, um objeto.

Criaremos uma classe `Jogo` para representar esses dados. Para resumirmos um pouco a criação, essa classe terá apenas o construtor e as variáveis dela (`nome`, `categoria` e `console`), que não serão inicializadas como privadas (para não perdermos tempo criando `properties`):

```
class Jogo:  
    def __init__(self, nome, categoria, console):  
        self.nome = nome  
        self.categoria = categoria  
        self.console = console
```

Lembre-se: é necessário criar variáveis privadas para controlar acesso aos seus atributos!

Mesmo que a nossa classe não esteja seguindo todos os padrões, já podemos usá-la. Observe o seguinte trecho do nosso código:

```
def ola():  
    lista = ['Tetris', 'Super Mario', 'Pokemon Gold']
```

Nele, estamos criando uma lista de strings, ou seja, de textos. Para criarmos uma lista de objetos que contêm características diferentes, a primeira coisa que precisaremos fazer é criar esses objetos:

```
def ola():  
    jogo1 = Jogo('Super Mario', 'Acao', 'SNES')  
    jogo2 = Jogo('Pokemon Gold', 'RPG', 'GBA')
```

Agora, criaremos a lista que contém esses dois jogos:

```
def ola():  
    jogo1 = Jogo('Super Mario', 'Acao', 'SNES')  
    jogo2 = Jogo('Pokemon Gold', 'RPG', 'GBA')  
    lista = [jogo1, jogo2]  
    return render_template('lista.html', titulo='Jogos',  
                           jogos=lista)  
  
app.run()
```

Vamos testar? Rodando a aplicação e acessando a URL <http://127.0.0.1:5000/inicio> (<http://127.0.0.1:5000/inicio>), teremos a seguinte resposta:

```
Jogos
```

```
Nome
```

```
<main.Jogo object at 0x00000297BE144588>
```

```
<main.Jogo object at 0x00000297BE1444E0>
```

Isso acontece porque estamos exibindo o conteúdo de `jogo`, e não o texto de cada atributo. Resolveremos esse problema adiante. Até lá!