

Para saber mais: Outra forma de representação

Nós vimos como usar `__str__` para representar um objeto como string de forma legível.

Falamos sobre uma outra forma de representação, ela pode ajudar bastante se precisarmos encontrar um erro, ou *debugar* o código.

Assim como o `__str__`, existe outro método especial chamado `__repr__`, que é usado para mostrar uma representação que ajude no *debug* ou *log* de um código.

Por exemplo, se você quiser entender como funciona seu objeto, ou se está válido, e imprimir o seu valor string, qual resultado abaixo facilita sua vida?

```
> Filme(nome='vingadores', ano=2018, duracao=160)
```

Ou

```
> "Filme: Vingadores de 2018 - 160 min"
```

A **primeira** deixa bem claro como funciona o objeto. Normalmente, a **segunda** forma ilustra o que um usuário final ficaria satisfeito em ver.

A ideia da primeira versão é remover ambiguidade e permite, por exemplo, recriar o objeto, já que está mostrando todas as informações.

Outro exemplo, se chamarmos o `repr` de um objeto do tipo `list`, podemos ter uma ideia do que é esperado quando criarmos o nosso próprio com `__repr__`:

```
lista = [1, 2, 4, 5]

print(repr(lista))
```

Se pegarmos o resultado do `print`, conseguimos recriar o objeto `lista`.