09

Para saber mais: Set

Quando uma sequência não é suficiente

Já aprendemos algumas características das sequências. Elas guardam valores de qualquer tipo de dado e possuem uma determinada ordem, pois possuem um índice. As sequências também são chamadas de coleções e são o *arroz e feijão* para qualquer desenvolvedor Python!

Agora dá uma olhada no exemplo abaixo onde criamos uma lista de CPFs:

```
lista = [11122233344, 22233344455, 33344455566]
```

Nada errado aqui, mas imagine agora que você precisa evitar que exista algum CPF duplicado nesta lista. Isso é algo muito comum no dia-a-dia! Em muitas circunstâncias precisamos de uma coleção que não permita valores duplicados, mas nada nos impede adicionar um mesmo CPF nessa lista, por exemplo:

```
lista.append(11122233344) #funciona!
```

Isso também funciona se fosse uma tuple! Uma tuple também permite elementos duplicados!

Conhecendo o set

E agora? Será que o Python não oferece alguma coleção onde não podem existir elementos duplicados? Claro que existe uma coleção com esse propósito e ela se chama **set**.

Veja o mesmo exemplo, mas agora inicializando um set:

```
colecao = {11122233344, 22233344455, 33344455566}
```

Repare que usamos `{}` chaves para declarar os elementos iniciais. Pouca diferença comparando com as sequências, não?

Adicionando elementos no set

Para adicionar um elemento a um set devemos chamar a função `add` (invés da `append`):

```
colecao.add(44455566677) #vai adicionar pois não existe ainda
```

E se usarmos um CPF que já existe? Não deve funcionar, certo? Vamos testar,e ver que o set vai ignorá-lo:

```
colecao.add(11122233344) #nao vai adicionar pois este CPF já existe!
```

set não possui um índice

É importante notar que o set não é uma sequência, pois não tem um índice. O código abaixo não funciona:

```
colecao[0]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'set' object does not support indexing
```

Isso mesmo, não tem índice. E como não temos um índice não sabemos em qual ordem vem os valores quando imprimimos um **set** de dentro de um laço `for`:

```
for cpf in colecao:
     print(cpf)
```

Imprime:

```
11122233344
44455566677
33344455566
22233344455
```

Repare que os elementos foram listados fora da ordem de inserção. Isso acontece porque o set não é ordenado.

Resumindo

Um set é uma coleção não ordenada de elementos. Cada elemento é único, isso significa que não existem elementos duplicados dentro do set.

Respire fundo e fique tranquilo pois o `set` será abordado ainda com mais detalhe em outros cursos. Vamos continuar?