**02 - Criando dicionários**

Neste capítulo conheceremos uma nova estrutura de dados utilizada em Python, e que você certamente usará em seus projetos de *data science*: os dicionários. Sabemos que listas são coleções sequenciais ordenadas que utilizam índices para acessar os seus valores. Os dicionários são coleções um pouco diferentes, pois são estruturas de dados que representam um tipo de mapeamento - coleções de associações entre pares de valores nas quais o primeiro elemento é conhecido como chave e o segundo como valor.

No notebook do curso temos um pequeno texto sobre o assunto, além da representação de um dicionário básico.

Listas são coleções sequenciais, isto é, os itens destas sequências estão ordenados e utilizam índices (números inteiros) para acessar os valores.

Os dicionários são coleções um pouco diferentes. São estruturas de dados que representam um tipo de mapeamento. Mapeamentos são coleções de associações entre pares de valores onde o primeiro elemento do par é conhecido como chave (key) e o segundo como valor (value).

dicionario = {key\_1: value\_1, key\_2: value\_2, ..., key\_n: value\_n} [Clicando aqui você pode acessar as referências na documentação do Python](https://docs.python.org/3.6/library/stdtypes.html#typesmapping).

Para entendermos as diferenças entre os dicionários e as listas, inicializaremos as variáveis carros e valores com os seguintes itens:

carros = ['Jetta Variant', 'Passat', 'Crossfox']

carrosCOPIAR CÓDIGO

valores = [88078.64, 106161.94, 72832.16]

valoresCOPIAR CÓDIGO

Como as litas (assim como as tuplas) são ordenadas, sabemos que o primeiro valor é referente ao veículo "Jetta Variant", o segundo ao "Passat" e o terceiro ao "Crossfox". Entretanto, seria mais simples trabalharmos esses valores se eles estivessem em uma única estrutura de dados - ainda que, em um conjunto com tão poucos dados, essa tarefa não pareça tão complexa. Se tivéssemos um exemplo com 200 itens, certamente seria mais desgastante trabalharmos com suas posições.

Conseguimos acessar o índice de um determinado elemento usando a função index(), como no exemplo abaixo:

carros.index('Passat')COPIAR CÓDIGO

1

Para acessarmos o valor desse carro na lista valores, precisaríamos de uma sintaxe que incluísse tal verificação, por exemplo:

valores[carros.index('Passat')COPIAR CÓDIGO

Mais complexo, não? Tal acesso seria simplificado se tivéssemos uma estrutura que fizesse o mapeamento entre carros e valores, facilitando também outros tipos de análise. É justamente isso que os dicionários nos proporcionam, além de trazerem a possibilidade de criarmos dataframes, algo que abordaremos mais adiante no curso.

Para criarmos um dicionário, inicializaremos uma variável qualquer, como dados, que receberá a abertura e fechamento de chaves.

dados = {}COPIAR CÓDIGO

Dentro das chaves podemos passar uma estrutura de informações em pares, nas quais o primeiro valor representa a chave desse par. No nosso caso, as chaves serão os nomes dos veículos e os valores serão os seus respectivos preços.

dados = {'Jetta Variant': 88078.64, 'Passat': 106161.94, 'Crossfox': 72832.16}

dadosCOPIAR CÓDIGO

Assim teremos a estrutura do nosso dicionário:

{'Crossfox': 72832.16, 'Jetta Variant': 88078.64, 'Passat': 106161.94}

Note que recebemos os valores em outra ordem, mas isso não faz diferença. Os dicionários em si não são ordenados, pois possuem um mapeamento por referência e não precisam ser acessados por índices, mas sim por chaves. Se fizermos type(dados), conseguiremos verificar o tipo dessa informação: "dict", que é o tipo referente aos dicionários.

type(dados)COPIAR CÓDIGO

dict

Terminamos o capítulo anterior falando sobre a função zip() e o empacotamento de dados.

list(zip(carros, valores))COPIAR CÓDIGO

[('Jetta Variant', 88078.64), ('Passat', 106161.94), ('Crossfox', 72832.16)]

Podemos passar o nosso iterador para a função nativa dict() de modo a transformar essas informações em um novo dicionário. Para testarmos, inicializaremos novamente a variável dados, dessa vez recebendo a chamada de dict(). Esta, por sua vez, receberá a chamada de zip() com os parâmetros carros e valores.

dados = dict(zip(carros, valores))

dadosCOPIAR CÓDIGO

{'Crossfox': 72832.16, 'Jetta Variant': 88078.64, 'Passat': 106161.94}

Essa é outra forma de criarmos dicionários, relativamente mais simples, já que utiliza somente as variáveis que já havíamos preparado antes. Assim como as listas, os dicionários são mutáveis e podem ser utilizados em operações. Esse será o assunto do nosso próximo vídeo!