**09 - Métodos de listas**

Conheceremos agora alguns métodos que nos permitem fazer modificações nas listas que estamos criando, como a adição de itens, a ordenar da lista, a remoção de itens e a cópia. Na seção **3.4 Métodos de listas** de nosso notebook, deixamos um link para a documentação do Python sobre esse assunto. É importante ter o hábito de consultar a documentação, pois podem existir pontos específicos do seu trabalho que não sejam resolvidos com os métodos apresentados aqui no curso, mas talvez existam outros métodos que possam auxiliar. Também é possível criar novas funções, mas esse é um assunto para o próximo curso.

Aqui trabalharemos com os métodos **sort()**, **append()**, **pop()** e **copy()**. Para facilitarmos os estudos, rodaremos novamente a definição da variável Acessorios.

Acessorios = ['Rodas de liga', 'Travas elétricas', 'Piloto automático', 'Bancos de couro', 'Ar condicionado', 'Sensor de estacionamento', 'Sensor crepuscular', 'Sensor de chuva']

AcessoriosCOPIAR CÓDIGO

['Rodas de liga', 'Travas elétricas', 'Piloto automático', 'Bancos de couro', 'Ar condicionado', 'Sensor de estacionamento', 'Sensor crepuscular', 'Sensor de chuva']

O método sort() funciona por meio da sintaxe A.sort(), onde A representa uma lista genérica. Note que a nossa lista não está ordenada seguindo alguma lógica, como a ordem alfabética. O sort() nos permite fazer isso, dentre outras possibilidades, como ordenar os valores dos carros. Esse tipo de ordenação nos permite fazer algumas análises preliminares.

Acessorios.sort()

AcessoriosCOPIAR CÓDIGO

O método sort() modifica a lista por meio da qual ele foi chamado, resultando, nesse caso, em uma lista de acessórios em ordem alfabética:

['Ar condicionado', 'Bancos de couro', 'Piloto automático', 'Rodas de liga', 'Sensor crepuscular', 'Sensor de chuva', 'Sensor de estacionamento', 'Travas elétricas']

O método append() adiciona um elemento passado como parâmetro ao final da lista a partir do qual foi chamado.

Acessorios.append('4 x 4')

AcessoriosCOPIAR CÓDIGO

['Ar condicionado', 'Bancos de couro', 'Piloto automático', 'Rodas de liga', 'Sensor crepuscular', 'Sensor de chuva', 'Sensor de estacionamento', 'Travas elétricas', '4 x 4']

Já o pop() faz exatamente o contrário. Se não passarmos nenhum parâmetro, ele removerá o último elemento da lista, retornando o elemento removido.

Acessorios.pop()COPIAR CÓDIGO

'4 x 4'

Se visualizarmos novamente a lista, perceberemos que o item 4 x 4 não será mais exibido.

AcessoriosCOPIAR CÓDIGO

['Ar condicionado', 'Bancos de couro', 'Piloto automático', 'Rodas de liga', 'Sensor crepuscular', 'Sensor de chuva', 'Sensor de estacionamento', 'Travas elétricas']

Também podemos remover um elemento específico da lista utilizando a indexação. Por exemplo, se quisermos remover Rodas de liga, teremos que passar como parâmetro o índice desse elemento, que é 3.

Acessorios.pop(3)COPIAR CÓDIGO

'Rodas de liga'

Com isso, nossa lista deixará de conter esse item.

AcessoriosCOPIAR CÓDIGO

['Ar condicionado', 'Bancos de couro', 'Piloto automático', 'Sensor crepuscular', 'Sensor de chuva', 'Sensor de estacionamento', 'Travas elétricas']

Imagine que desejamos fazer alterações drásticas em uma lista para realizarmos testes, mas ainda mantendo o conteúdo da original. O método copy() pode ser usado para fazer uma cópia de uma lista. Porém, você deve imaginar que também é possível fazer isso criando uma nova variável e atribuindo a ela o conteúdo da primeira, como no exemplo abaixo:

Acessorios\_2 = Acessorios

Acessorios\_2COPIAR CÓDIGO

['Rodas de liga', 'Travas elétricas', 'Piloto automático', 'Bancos de couro', 'Ar condicionado', 'Sensor de estacionamento', 'Sensor crepuscular', 'Sensor de chuva']

Feito isso, adicionaremos o elemento "4 x 4" à lista Acessorios\_2 usando o método append().

Acessorios\_2.append('4 x 4')

Acessorios\_2COPIAR CÓDIGO

['Rodas de liga', 'Travas elétricas', 'Piloto automático', 'Bancos de couro', 'Ar condicionado', 'Sensor de estacionamento', 'Sensor crepuscular', 'Sensor de chuva', '4 x 4']

A ideia é que a lista Acessorios mantenha-se a mesma, sem a adição do novo item, certo? Porém, observe o que acontece quando exibimos o seu conteúdo:

AcessoriosCOPIAR CÓDIGO

['Rodas de liga', 'Travas elétricas', 'Piloto automático', 'Bancos de couro', 'Ar condicionado', 'Sensor de estacionamento', 'Sensor crepuscular', 'Sensor de chuva', '4 x 4']

Repare que a lista Acessorios também foi modificada. Isso acontece porque a instrução Acessorios\_2 = Acessorios apenas dá um novo rótulo a uma referência armazenada na memória. Sendo assim, tudo alteração feita a essa referência usando Acessorios\_2 modificará também a lista Acessorios. Há o método copy() não copia somente a referência, criando uma nova área na memória para a lista a partir do qual foi chamado.

Para testarmos, chamaremos Acessorios.pop() de modo a removermos o item "4 x 4". Obviamente, se chamarmos Acessorios\_2, o tem também terá sido removido. Agora criaremos uma nova lista Acessorios\_2 que receberá Acessorios.copy().

Acessorios\_2 = Acessorios.copy()

Acessorios\_2COPIAR CÓDIGO

['Rodas de liga', 'Travas elétricas', 'Piloto automático', 'Bancos de couro', 'Ar condicionado', 'Sensor de estacionamento', 'Sensor crepuscular', 'Sensor de chuva']

A partir dessa lista, chamaremos novamente o método append() passando como parâmetro o item "4 x 4", o que fará com que esse elemento seja listado.

Acessorios\_2.append('4 x 4')

Acessorios\_2COPIAR CÓDIGO

['Rodas de liga', 'Travas elétricas', 'Piloto automático', 'Bancos de couro', 'Ar condicionado', 'Sensor de estacionamento', 'Sensor crepuscular', 'Sensor de chuva', '4 x 4']

Entretanto, ele não será exibido na lista Acessorios, já que elas são independentes.

AcessoriosCOPIAR CÓDIGO

['Rodas de liga', 'Travas elétricas', 'Piloto automático', 'Bancos de couro', 'Ar condicionado', 'Sensor de estacionamento', 'Sensor crepuscular', 'Sensor de chuva']

Outra maneira de realizarmos uma cópia atribuindo um novo lugar na memória é por meio da sintaxe **A[:]**, onde A representa uma lista qualquer. Observe o exemplo:

Acessorios\_2 = Acessorios[:]COPIAR CÓDIGO

Essa sintaxe é equivalente ao método copy() - ou seja, ambos têm a mesma funcionalidade. No próximo capítulo conversaremos sobre estruturas de repetição e condicionais.