**07 - Instrução if**

Nosso assunto agora é a instrução **if**, com a qual conseguimos testar uma condição e executar um código caso ela seja verdadeira, seguindo o formato padrão abaixo:

if <condição>:

<instruções caso a condição seja verdadeira>COPIAR CÓDIGO

Ou seja, se (if) uma <condição> for verdadeira, executaremos as <instruções caso a condição seja verdadeira>. Note que a linha em que usamos o if deve ser terminada com dois pontos (:), e a instrução a ser executada deve ser precedida por uma indentação.

Outro recurso interessante de ser abordado nesse momento são os **operadores de comparação**. Já trabalhamos com operadores de atribuição, por exemplo, quando atribuímos um valor a uma variável (x = 2), rotulando um espaço na memória onde aquele valor é armazenado.

Nesse caso, queremos realizar comparações, como testar de x é igual a 2. Nesse caso, usaríamos o operador de comparação ==.

== igual a

!= diferente de

> maior que

< menor que

>= maior igual a

<= menor igual a COPIAR CÓDIGO

Existem também os **operadores lógicos** **and**, **or** e **not**, com os quais trabalharemos no futuro. Para trabalharmos com o if, deixamos preparada uma lista dados contendo sublistas com as informações de alguns veículos - nome, ano de fabricação e se o veículo é ou não zero km.

# 1º item da lista - Nome do veículo

# 2º item da lista - Ano de fabricação

# 3º item da lista - Veículo é zero km?

dados = [

['Jetta Variant', 2003, False],

['Passat', 1991, False],

['Crossfox', 1990, False],

['DS5', 2019, True],

['Aston Martin DB4', 2006, False],

['Palio Weekend', 2012, False],

['A5', 2019, True],

['Série 3 Cabrio', 2009, False],

['Dodge Jorney', 2019, False],

['Carens', 2011, False]

]

dadosCOPIAR CÓDIGO

Primeiramente usaremos o for para iterarmos por dados chamando cada elemento de lista. Se fizermos um print(list), teremos como retorno a varredura de cada uma das sublistas.

for lista in dados:

print(lista)COPIAR CÓDIGO

['Jetta Variant', 2003, False] ['Passat', 1991, False] ['Crossfox', 1990, False] ['DS5', 2019, True] ['Aston Martin DB4', 2006, False] ['Palio Weekend', 2012, False] ['A5', 2019, True] ['Série 3 Cabrio', 2009, False] ['Dodge Jorney', 2019, False] ['Carens', 2011, False]

A ideia agora é imprimirmos somente os veículos marcados como True, ou seja, os que são zero km. Para isso, dentro do for, incluiremos um if e abriremos um par de parênteses (()). Podemos acessar o último valor de cada índice por meio do seu índice - como se trata do terceiro valor, o índice é 2. Como exemplo, podemos simplesmente imprimir todos os valores com esse índice em lista.

for lista in dados:

print(lista[2])COPIAR CÓDIGO

False False False True False False True False False False

Com isso, conseguimos acessar o valor que indica se um carro é zero km ou não. Agora incluiremos, no laço for, um if que verifica se a posição list[2] é igual (==) a True. Em caso positivo, imprimiremos a lista.

for lista in dados:

if (lista[2] == True):

print(lista)COPIAR CÓDIGO

Como retorno teremos somente os veículos zero km desse conjunto.

['DS5', 2019, True] ['A5', 2019, True]

Agora temos a possibilidade de criar um laço e, dentro dele, verificar uma condição, executando determinada instrução caso a condição passe. Para tornarmos nosso código um pouco mais elegante, criaremos uma lista vazia zero\_km\_Y. Caso a verificação retorne verdadeiro, adicionaremos o item lista à nova variável. Por fim, imprimiremos o seu conteúdo.

zero\_km\_Y = []

for lista in dados:

if (lista[2] == True):

zero\_km\_Y.append(lista)

zero\_km\_YCOPIAR CÓDIGO

[['DS5', 2019, True], ['A5', 2019, True]]

Como ainda não conhecemos as instruções else e elif, podemos construir outro código para a situação contrária. No caso, teremos uma variável zero\_km\_N e verificaremos de lista[2] retorna False.

zero\_km\_N = []

for lista in dados:

if (lista[2] == False):

zero\_km\_N.append(lista)

zero\_km\_NCOPIAR CÓDIGO

[['Jetta Variant', 2003, False], ['Passat', 1991, False], ['Crossfox', 1990, False], ['Aston Martin DB4', 2006, False], ['Palio Weekend', 2012, False], ['Série 3 Cabrio', 2009, False], ['Dodge Jorney', 2019, False], ['Carens', 2011, False]]

Assim conseguimos criar agrupamentos diferentes, o que nos permitirá testar coisas diferentes - por exemplo, um modelo somente para veículos 0 km ou veículos que não são 0 km.

Também é possível executarmos esse código por meio das *list comprehensions*. Como o resultado final é uma lista, esse será o primeiro parâmetro que passaremos entre colchetes. Em seguida, incluiremos o iterador for, varrendo a lista dados, e então a instrução if.

[lista for lista in dados if lista[2] == True]COPIAR CÓDIGO

[['DS5', 2019, True], ['A5', 2019, True]]

Com isso replicamos o código anterior, agora usando uma única linha. No próximo vídeo usaremos o else para executarmos código também quando uma condição não é verdadeira.