**06 - Definindo funções que retornam valores**

Nesse vídeo solucionaremos o problema que identificamos no vídeo anterior: construímos uma função que não retorna valores e, por isso, não conseguimos atribuir o seu resultado a uma variável para utilizá-lo no código posteriormente. O formato padrão de uma função que retorna um valor não é muito diferente daquele que já estamos acostumados.

def <nome>(<param\_1>, <param\_2>, ..., <param\_n>):

<instruções>

return <resultado>COPIAR CÓDIGO

A diferença básica é que, após as instruções, temos a palavra-chave return seguida do valor que gostaríamos de retornar (aqui chamado de "resultado"). É possível retornar um ou mais valores, e começaremos com uma função que retorna um único valor. A última função que criamos no vídeo anterior calculava a média dos valores de uma lista e o imprimia na tela.

def media(lista):

valor = sum(lista) / len(lista)

print(valor)COPIAR CÓDIGO

Essa é uma função bastante simplificada, apenas para entendermos a estrutura e o funcionamento desse recurso. No dia-a-dia, é interessante manter uma atenção mais aprimorada, testando se os dados estão fazendo sentido e coisas do tipo.

Mantendo a estrutura dessa função, ao invés de imprimirmos o valor, vamos retorná-lo com a palavra-chave return.

def media(lista):

valor = sum(lista) / len(lista)

return valorCOPIAR CÓDIGO

O resultado final será o mesmo, calculando a média de todos os valores contidos na lista passada por parâmetro.

media([1, 2, 3, 4, 5, 6, 7, 8])COPIAR CÓDIGO

4.5

A diferença é que agora podemos atribuir a chamada de media() a uma variável, que chamaremos de resultado. Se exibirmos o seu conteúdo, receberemos exatamente a informação devolvida pela função.

resultado = media([1, 2, 3, 4, 5, 6, 7, 8])

resultadoCOPIAR CÓDIGO

4.5

Como dito anteriormente, uma função pode retornar mais de um valor. Em nosso exemplo continuaremos mantendo a mesma estrutura da função media(), mas, ao invés de retornarmos somente o valor, devolveremos também o número de elementos utilizados para esse cálculo - ou seja, o tamanho da lista (len(lista)), ambos contidos em uma tupla.

def media(lista):

valor = sum(lista) / len(lista)

return (valor, len(lista))COPIAR CÓDIGO

Para testarmos, executaremos o mesmo cálculo de média que fizemos acima.

resultado = media([1, 2, 3, 4, 5, 6, 7, 8])

resultadoCOPIAR CÓDIGO

Dessa vez nosso retorno será uma tupla contendo dois valores.

(4.5, 8)

Se aumentarmos o número de elementos para 9, os valores (média e tamanho da lista) serão alterados.

resultado = media([1, 2, 3, 4, 5, 6, 7, 8, 9])

resultadoCOPIAR CÓDIGO

(5.0, 9)

Nesse curso aprendemos o conceito de desempacotamento de tuplas, o que nos permitia atribuir valores diferentes de uma tupla a variáveis diferentes. Isso também é possível com o retorno de funções. Ou seja, podemos criar, na mesma chamada, as variáveis resultado e n, esta última representando o tamanho da lista.

resultado, n = media([1, 2, 3, 4, 5, 6, 7, 8, 9])

resultadoCOPIAR CÓDIGO

5.0

nCOPIAR CÓDIGO

9

Com isso encerramos nosso conteúdo sobre funções. Ao longo desse curso básico de Python você adquiriu bagagem o suficiente para iniciar e desenvolver seus trabalhos em *data science*. Na próxima seção do treinamento, faremos uma pequena introdução à biblioteca Pandas, também muito importante na ciência de dados.