02

Upper e Lower

Transcrição

[00:00] Até agora, retornamos os argumentos moeda origem e moeda destino. Mas além disso, precisamos também retornar o argumento valor, que nem inserimos na url ainda, mas iremos inserir daqui a pouco.

[00:15] Antes disso, quero propor para vocês um erro novo. A aplicação do ByteBank tem vários bugs, e nossa classe tem que estar pronta para isso. Supondo que algum nome de argumento venha com qualquer caractere maiúsculo, será que vai ter algum problema? Será que o Python sabe a diferença de maiúsculo e minúsculo? Vamos ver.

[00:40] Vou criar duas variáveis: banco1 = “bytebank” e banco2 = “Bytebank”. Vou perguntar para o Python se banco1 é igual a banco2. Ele disse que não. Banco1 não é igual a banco2. O Python sabe a diferença entre letra maiúscula e minúscula.

[01:16] Vamos supor que o r de origem venha maiúsculo na nossa url. Minha moeda destino continua normal, mas minha moeda origem não veio. Então, com certeza, um erro desse tipo gera um problema grande dentro da nossa publicação.

[01:45] O Python possui duas funções bem legais, dois métodos string. O upper e o lower. Um deixa todos os caracteres de uma string em maiúscula e o outro deixa todos em minúscula. Vou fazer o seguinte: banco2 = “Bytebank”.upper () e vou printar. Ele deixou todos os caracteres em maiúsculo. O contrário também é valido se eu colocar lower.

[02:25] Esses dois métodos podem nos ajudar muito a não ter esse tipo de problema na nossa classe. O que temos que fazer é pegar essa lógica e jogar para dentro dela. Onde seria interessante fazer isso e qual dos dois eu faço? Isso é só uma definição. Eu quero que minha função trabalhe só com letras minúsculas ou só com maiúsculas? Acredito que seja uma boa usar somente letras minúsculas. Então, no momento em que recebo esse atributo, já faço a transformação. buscaMoedaOrigem = “moedaorigem”.lower () buscaMoedaDestino = “moedadestino”.lower ()

[03:15] Assim, não tem risco nenhum da minha url ter alguma coisa maiúscula. Dólar e real foram retornados perfeitamente.

[03:38] O que podemos fazer agora é já encontrar o valor, aproveitando que estamos mexendo na classe. Eu vou terminar essa url, vou colocar o último argumento dela: url = “https://bytebank.com/cambio?moedaorigem=moedadestino&moedadestino=dólar&valor=1500” (https://bytebank.com/cambio?moedaorigem=moedadestino&moedadestino=d%C3%B3lar&valor=1500%E2%80%9D)

[03:50] Preciso fazer uma coisa bem parecida com ExtraiArgumentos. Eu vou criar um ExtraiValor. Novamente vou criar uma variável que vai me ajudar a buscar alguma coisa. Eu vou usar isso para me ajudar a localizar o índice inicial. E preciso retornar o índice dele mais o comprimento. Temos um método já bem interessante para isso, que é o encontraIndiceInicial. def extraiValor (self): buscaValor = “valor=” indiceInicialValor = self.encontraIndiceInicial (buscaValor) valor = self.url [indiceInicialValor:] return valor

[05:11] Rodando isso, ele me deu dólar imprimindo todo o resto. Eu quero só o dólar na moeda destino. Esse problema apareceu por quê? Eu estou deixando o último índice do fatiamento da moeda destino vazio. Eu encontro o inicial, mas não encontro o final, porque antes a minha url acabava ali, agora não mais.

[05:47] Eu preciso encontrar o índice final da moeda destino e passá-lo para o fatiamento. Mas qual pode ser esse índice? Posso encontrar o &valor, passar esse índice. Testando, o bug fica corrigido.

[06:20] O que eu preciso agora é criar uma variável nova valor = argumentosUrl.extraiValor (). Printando essa variável, estou retornando meu dólar, meu real e meu valor. Nossa classe está completa, está suportando um bug do valor da moeda origem

vir incorreto, o bug de vir algum caractere no nome dos argumentos em maiúsculo, e está retornando tudo o que eu quero.

[07:15] Na próxima aula, quero propor que caso a url não seja do ByteBank, será que minha classe está pronta? Como fazemos para identificar se a url pertence ao Byte Bank ou não? Vejo vocês lá.