03

O método Len

Transcrição

[00:00] Nossa ideia agora é fortalecer mais nossa classe. Tivemos um problema, que foi encontrar o igual duas vezes. Quero mostrar um pouco melhor como funciona o find, porque a ideia agora é, ao invés de buscar o sinal de igual, buscar o nome do argumento.

[00:22] Antes disso, quero mostrar para vocês como o Python se comporta quando passamos um texto maior, ao invés de um só caractere, porque até agora só passamos o & e o sinal de igual, que são caracteres específicos. Mas e se eu passar "moedaDestino"? Será que ele me retorna o índice no O? No M? Vamos dar uma olhada: index = url.find('"moedaDestino") print (index)

[01:15] Ele me dá o 17. Se contarmos, é exatamente onde queríamos. Vamos colocar agora a subString. index = url.find('"moedaDestino") subString = url [index:] print (subString)

[01:50] Isso me gera um problema. Eu preciso começar no índice do D, não no índice do M. Mas perceba que o que tem entre esse índice que encontrei com o find tem a quantidade de caracteres do nome do meu argumento. Depois, é só somar +1. Se eu colocar index = url.find('"moedaDestino") + 12, ele me dá =dólar. Se eu colocar index = url.find('"moedaDestino") + 13, ele me dá dólar.

[02:35] Mas e se eu mudar o nome do meu argumento em alguma ocasião? Em algum ponto tenho que mudar esse nome. Será que preciso ficar mexendo na linha de código o tempo todo? Ou será que o Python tem uma função específica que serve para me retornar o tamanho de uma string? Sim, o nome desse método é len.

[02:56] Ele não é um método de string. É uma função do Python index = url.find('"moedaDestino") +len("moedaDestino")

[03:18] Assim, ele chega no =dólar. Se eu colocar +1, ele me dá dólar. Perfeito. Retorno já o valor do argumento que eu quero para dentro do meu código. Agora, o que precisamos fazer é pegar essa lógica e jogar para dentro da nossa classe. Vou criar mais duas variáveis: buscaMoedaOrigem = "moedaorigem" buscaMoedaDestino = "moedaDestino"

[03:56] Vou usar isso para alimentar meu método find e meu método len. Eu preciso pegar a instância, que é o url que estou passando, e preciso usar o método find nela. Perceba que para o moeda origem posso fazer a mesma coisa, só vou mudar a variável. indiceInicialMoedaDestino = self.url.find(buscaMoedaDestino) + len (buscaMoedaDestino) + 1 indiceInicialMoedaOrigem = self.url.find(buscaMoedaOrigem) + len (buscaMoedaOrigem) + 1 indiceFinalMoedaOrigem = self.url.find("&")

[05:48] Rodando esse código, ele me dá dólar e real. Funciona normalmente.

[06:09] Agora, o teste é: se eu colocar todo aquele sufixo antes que representa o endereço da minha página, será que vai continuar funcionando normalmente? url = "https://bytebank.com/cambio?moedaorigem=real&moedadestino=dólar" (https://bytebank.com/cambio?moedaorigem=real&moedadestino=d%C3%B3lar%E2%80%9D)

[06:28] Funciona normalmente. Agora não tenho mais aquele problema de encontrar o igual de um ou de outro. Eu busco direto no nome do argumento.

[06:40] Percebam que eu fiz isso duas vezes, e se eu fiz duas vezes, acho que é uma justificativa para criar um método que vai ser responsável por encontrar esse índice. Eu preciso trazer aquela lógica lá de cima para esse método. def encontraIndiceInicial (self, moedaBuscada): return self.url.find(moedaBuscada) + len (moedaBuscada) + 1

[07:35] Aí, ao invés de passar a lógica inteira, eu chamo minha função: indiceInicialMoedaDestino = self.encontraIndiceInicial (buscaMoedaDestino) indiceInicialMoedaOrigem = self.encontraIndiceInicial (buscaMoedaOrigem)

[08:00] Testando novamente, tudo certo. Retornei minha moeda destino e minha moeda origem.

[08:16] Por enquanto é só. Vejo vocês na próxima aula.