

3.2 - Nosso primeiro algoritmo (parte 1)

00;00;00;26 - 00;00;20;22

Olá pessoal! Chegou o esperado momento. Na aula de hoje nós vamos fazer nosso primeiro programa em Python. Bora lá! Mas galera, sabe o que rolou? Me deu uma vontade de comer um bolo de cenoura, sabe? Deixa eu ver o que eu preciso fazer aqui. Bom, eu preciso fazer uma lista com os ingredientes, depois eu preciso ir no mercado comprar.

00;00;20;22 - 00;00;40;17

No mercado eu preciso verificar se tem todos os ingredientes que eu preciso lá e, caso contrário, eu vou ter que substituir por algo parecido. Vocês perceberam o que eu fiz? Eu escrevi passo a passo em pequenas instruções o que eu preciso fazer. Acabei de fazer um algoritmo. Isso mesmo. O algoritmo é um conjunto de instruções e regras para executar alguma função ou tarefa.

00;00;41;05 - 00;01;09;00

Agora vamos pegar esse algoritmo aí de comprar os ingredientes para fazer o bolo e deixar ele de uma forma mais visual. Vamos fazer um fluxograma. Aqui nós temos o início do nosso algoritmo sinalizando aqui dentro dessa caixinha escrito "início". O próximo passo é fazer a lista. Então dentro desse retângulo eu coloco "fazer a lista de ingredientes". Depois de ter a lista, eu preciso ir no mercado. Então eu já coloco aqui nesse próximo retângulo a instrução de ir ao mercado.

00;01;09;17 - 00;01;33;16

Essas formas estão ligadas por setas que indicam qual o caminho que vai seguir. Por exemplo, a seta vai de fazer a lista para depois ir ao mercado e não o contrário. Então devemos nos atentar a essa ordem dessas instruções para não ter nenhuma confusão. Beleza, cheguei no mercado. O próximo passo é procurar os ingredientes. Aqui, ao invés de colocar isso num retângulo, eu vou expandir em várias instruções para ficar mais fácil de entender a minha tarefa total.

00;01;33;27 - 00;01;58;19

Então, beleza, cheguei no mercado. Primeiro vou procurar se tem ovo. Ok. Veja bem, "se tem ovo" é uma condicional. Então a forma dessa instrução vai ser diferente: ao invés de um retângulo, vai ser um losango. Toda vez que a gente tiver uma condicional, vamos indicar por um losango. Dentro desse losango, eu vou colocar "tem ovo?". Caso positivo, eu vou para o próximo item da lista e no novo losango eu coloco "tem cenoura?".

00;01;59;12 - 00;02;21;22

E assim vai até o último item da lista. Chegou o último item, significa que colocamos tudo no carrinho, então as próximas ações são: somar a compra, pagar (importante) e levar para casa. Colocamos da mesma forma que usaram lá para indicar o início, mas agora a gente escreve o fim. Ok, olha que legal, nós aprendemos o que é um algoritmo.

00;02;21;22 - 00;02;42;20

Fizemos o algoritmo e ainda conseguimos fazer um fluxograma no nosso algoritmo. E digo mais: nesse caso, a tarefa aqui foi comprar os ingredientes para fazer um bolo, mas qualquer tarefa do dia a dia pode ser transformada em um algoritmo. Você pode fazer um algoritmo para ir ao trabalho. Você acorda, escova os dentes, toma café, etc. Que tal vocês fazerem um algoritmo assim e compartilhar?

00;02;43;03 - 00;03;05;07

Fica aí de desafio. E, gente, eu tinha falado que se eu fosse no mercado e não encontrasse algum dos ingredientes, eu ia procurar alguma substituição, né? Tipo, particularmente, não sei se tem uma substituição pra fermento, então se a resposta do “tem fermento?” é “não”, eu vou direto por fim. Mas, por exemplo: “tem leite?”. Se não tiver leite de vaca, eu posso tentar uma substituição.

00;03;05;24 - 00;03;21;12

Então “tem leite de vaca?”. Se a resposta for “não”, “tem leite de soja?”, que é uma substituição. Se for “sim”, eu sigo a minha lista e continuo lá: se tem óleo, se tem fermento. Agora, se não tem nem o leite de vaca e nem leite de soja, eu já vou para o final, porque provavelmente eu não vou ter uma substituição.

00;03;22;01 - 00;03;44;10

É isso o nosso fluxograma. Agora sim, mãos no teclado. Vamos pegar esse algoritmo e transformar em um código Python. Bom, vamos elaborar um código para fazer a soma dos preços dos ingredientes pra gente, a soma de apenas os itens que têm mercado. Assim, a gente tem uma ideia de quanto a gente vai gastar e tal. Vamos lá fazer o script...

00;03;44;12 - 00;04;18;11

Bom, como nesse nosso script a ideia é fazer a etapa de soma dos preços dos ingredientes, nós precisamos criar variáveis com os preços. Nós vimos na última aula o que são variáveis, mas precisamos saber que existem algumas regrinhas na hora de criar o nome das variáveis e que as nossas variáveis podem ser de diferentes tipos. Mas vamos primeiro aos nomes. Por exemplo, vou abrir aqui o meu bloco de notas que é onde a gente vai fazer o nosso código e vamos colocar aqui um nome: preço cenoura.

00;04;18;11 - 00;04;41;29

Pro preço da cenoura, em Python, o padrão são as letras minúsculas e as palavras são separadas por underline. É aquela barrinha embaixo. Então, já comecei aqui errado. Coloquei tudo junto. Vou acrescentar aqui a minha barrinha para separar as duas palavras. Também não é indicado acentuação. Por isso, sem o “ç” cedilha, também já tinha colocado aqui. Vou substituir o “ç” por “c” e vamos fazer isso para os outros itens também.

00;04;41;29 - 00;05;31;12

Então a gente vai colocar: `preco_oleo`, `preco_fermento`, `preco_leite`, `preco_acucar`. Ok. Cenoura, óleo, fermento, leite, açúcar. Acho que são esses que a gente precisa para fazer o nosso bolo. Coloquei aqui em cada linha o nome de uma variável, mas para ela ser devidamente utilizada, eu preciso atribuir algum valor. E aí a gente precisa entender que as variáveis, elas podem receber diferentes tipos de valores. Aqui nessa tabela podemos ver alguns tipos de dados que podemos armazenar em variáveis.

00;05;32;02 - 00;05;59;28

Para variáveis numéricas, nós usamos os números inteiros (`int`). Aqui na tabela é a primeira opção: inteiro (`int`). Ou números com ponto flutuante também, aqueles números quebradinhos, decimais, como `float`. Temos também variáveis lógicas, no caso, que armazena a variável se é verdadeira ou falsa: `true` ou `false`. Elas são chamadas de booleanas. Aqui, o `bool`, booleano. A gente também tem variáveis de texto, que são as strings aqui embaixo: `string`.

00;05;59;29 - 00;06;27;23

Vocês podem ver na tabela que existem outros tipos: lista, tupla, dicionário, número complexo, mas pra gente o importante aqui nessa aula é entender o inteiro, o booleano e o string. Ok. Voltamos lá pro nosso script. Então aqui no preço cenoura eu posso colocar o sinal de igual e colocar o valor de 4.5, por exemplo.

00;06;27;23 - 00;07;02;11

Ah, nos números de ponto flutuante o que a gente indica o ponto flutuante é com o ponto, não com vírgula, beleza? Vamos atribuir alguns valores aqui aleatórios para os outros itens. Será completamente aleatório mesmo, tá, gente? Meu Deus, Carol, que roubo, que mercado é esse que você está pretendendo ir? Não sei. Ok, prontinho, criamos nossas variáveis e atribuímos aí alguns valores. Tendo ali os preços dos ingredientes, agora nós vamos verificar se eu tenho ingrediente na minha lista e, caso positivo, somar esse valor ao meu total.

00;07;02;25 - 00;07;27;26

Para isso, a gente vai usar uma função. Em programação, função ou método é um bloco de código que realiza uma tarefa específica. Para definir essa função, nós usamos a palavra chave `def`, de definition, seguido pelo nome da função em parêntese. Então vamos lá criar nossa função `def`. Para ser bem uma coisa bem elaborada, bem diferente, eu vou chamar de `soma_ingrediente`.

00;07;30;21 - 00;07;50;23

Colocamos dois pontos no final, que é para dizer que abaixo vai ter as instruções dessa função. E assim como no `if`, precisamos colocar tudo que estiver dentro da função com quatro espaços, a tabulação. Nossa função vai somar os preços de ingredientes que têm no mercado. Para isso, ela precisa saber, ou seja, receber a informação de algum jeito de quais ingredientes tem ou não.

00;07;51;16 - 00;08;31;25

Para isso, a gente usa parâmetros que são variáveis de entrada de uma função. Então vamos usar como parâmetros um: tem cenoura, tem açúcar, tem ovos etc. Onde indicamos que esses são os parâmetros da função? Basta colocar dentro dos parênteses da definição da função. Assim (tem_cenoura) vai receber se tem cenoura ou não, (tem_acucar) para açúcar e (tem_ovos), (tem_oleo), (tem_fermento), (tem_leite).

00;08;33;12 - 00;08;59;20

Então beleza. Colocamos os parâmetros, lembrando que os parâmetros eles precisam ser separados por vírgula. Então: tem_cenoura, tem_acucar e tal. Vamos dar uma olhada só para ver se está faltando alguma coisa. É importante a gente sempre revisar o código. Já vi aqui que coloquei “ç”. Não pode, é “c”. Então é “preco_fermento”. E aqui está faltando o que para fazer o nosso bolo? Está faltando o preço dos ovos, sem ovo a gente não vai conseguir fazer o nosso bolo.

00;08;59;24 - 00;09;25;09

Vou colocar aí 12 reais aí o preço dos ovos. Agora precisamos perguntar o que tem nesses parâmetros para ir somando. Antes, precisamos de uma nova variável para ir guardando a soma da nossa compra. Como ainda não vimos o que tem, vamos começar ela com o zero. Então vamos lá: dentro da nossa função soma_ingrediente... Deixa eu aumentar que eu acho que vai ficar melhor pra gente ver. Ó, ficou legal.

00;09;25;16 - 00;09;56;06

Dentro da nossa função soma_ingrediente, vou dar ali 4 espaços e aí eu vou colocar aqui uma variável total_compra igual a zero. Estou começando essa variável, atribuindo o valor de zero. Vamos começar verificando se tem cenoura. Pra isso, vamos fazer uma condicional. Então, continuando nossa função, podemos colocar assim aqui embaixo: total_compra, quatro espaços, if tem_cenoura, dois pontos. Sempre que tiver dois pontos indicando instruções relacionadas a uma função, condicional, etc,

00;09;56;17 - 00;10;23;14

Nós colocamos a indentação para ficar mais organizado e fácil de ler o código. Voltando: se tiver cenoura na nossa lista, ou seja, se “tem_cenoura” for igual a verdadeiro, no caso “true”, o que a gente vai fazer? Vamos somar o valor da cenoura na nossa variável total_compra. Apesar de saber que ela está zerada no começo, somamos com ela mesmo pra deixar o código organizado nos próximos ingredientes, pois caso tenha cenoura, ela já vai valer 4.5, que é o valor que a gente atribuiu para a cenoura.

00;10;23;14 - 00;10;52;23

E eu preciso somar esses 4.5 ao valor dos outros ingredientes que tiver. Então fica assim: aqui embaixo a gente dá os quatro espaços que é da função e mais quatro espaços que é do if. E aqui dentro a gente coloca total_compra igual a total_compra, porque a gente vai fazer acumulativo, mais preco_cenoura.

00;10;52;23 - 00;11;28;05

Verificando se está tudo certinho, vamos para o próximo ingrediente que vai fazer mais sentido. Vamos começar agora com o `tem_acucar`. Então vai ser uma próxima condicional. Aqui, dentro da função `soma_ingrediente`, quatro espaços, `if tem_acucar`, e embaixo quatro espaços dentro do `if` também, `total_compra` igual a `total_compra` mais preço do açúcar. Agora aqui a gente está somando. Vamos lá do início de novo...

00;11;28;06 - 00;11;51;13

Meu `total_compra` começa em zero. Se tiver cenoura na lista, vamos somar zero mais o preço da cenoura. Beleza? Então o `total_compra` vai ser igual ao preço da cenoura. Ok, próximo ingrediente... Se tiver açúcar, o `total_compra` vai ser o `total_compra` anterior, que era o preço da cenoura, mais o preço do açúcar. Percebeu que está acumulativo?

00;11;51;18 - 00;12;15;05

À medida que a gente vai avançando, os ingredientes vão somando. Por isso, a gente tem que somar o `total_compra` de novo para resgatar os valores dos ingredientes que já estavam na lista anteriormente. Então vamos fazer as condicionais para os outros ingredientes também. Bora lá. Agora, para retornar o valor, usamos o comando `return`. O `return` está dentro da função `def`, da função `def soma_ingrediente`.

00;12;15;15 - 00;12;40;21

Então vai ser só quatro espaços: `return total_compra`. O `return` não vai imprimir o nosso resultado. Por isso que a gente vai ter que usar o `print` pra poder printar esse `total_compra`, ok? Ele serve apenas para retornar esse resultado. Mas a gente precisa, antes de ir pra essa parte de printar e tal, precisamos chamar essa função e passar quais ingredientes temos.

00;12;41;08 - 00;13;05;14

Vamos começar tendo todos os ingredientes, então fora da função, agora fora dessa indentação que a gente tava colocando, a gente vai chamar a função e vamos colocar assim: `soma_ingredientes`. E como inicialmente a gente quer que tenha todos os ingredientes, vamos colocar tudo `true`, ou seja, como tudo verdadeiro. Carol, qual a quantidade de `true` que eu coloco aqui?

00;13;05;14 - 00;13;32;10

A mesma quantidade dos parâmetros que tem aqui no `soma_ingrediente`. Então a gente tem cenoura, açúcar, ovos, óleo, fermento e leite. Então vamos colocar 6 `true`. Um, dois, três, quatro, cinco, seis. E qual que eu sei que é o `true` correspondente a açúcar, a ovos? Enfim, vamos mostrar aqui, olha! Aqui nesse slide fica mais bonitinho da gente ver porque está por cores.

00;13;32;27 - 00;14;04;09

O primeiro true aqui é correspondente ao primeiro parâmetro da nossa função soma_ingredientes. Então se eu colocar aqui para false, por exemplo, eu estou querendo dizer que não tem cenoura, correspondente a cenoura. O segundo true é correspondente ao segundo elemento da nossa função soma_ingredientes. Então ao segundo, que é o açúcar. E por aí vai... Então, a gente tem que ficar muito atento à ordem dos parâmetros na função, porque essa ordem vai definir como que vai chegar os nossos valores para função.

00;14;04;23 - 00;14;40;10

Ok? Então, voltando aqui, a gente colocou tudo true inicialmente. Agora a gente vai printar o valor. Na verdade, como a gente colocou o return total_compra, quando a gente chamar essa função, ela vai retornar o valor dessa soma. Então vamos atribuir esse retorno dessa função a algum valor. Vamos colocar aqui "total". Então o total é igual ao quê? É igual ao total_compra, que é o que vai retornar dessa função. E agora sim, a gente pode printar aqui o nosso valor: print(total).

00;14;40;10 - 00;15;12;17

Ok, vamos salvar bonitinho. Vamos salvar como soma_ingredientes, porque a gente está muito criativo, ".py" porque é um arquivo .py. Lembrando aqui de colocar todos os arquivos, para não ser arquivos .txt e salvar. Pra executar, a gente vai lá abrir a nossa telinha preta, clicando em Windows. Aí tem aquelas várias formas, aí eu vou colocar dessa vez "terminal", vai aparecer lá prompt de comando.

00;15;14;03 - 00;15;41;26

A gente tem que ir lá na pasta onde a gente salvou esse .py, usando aquele comandinho de terminal "cd" para mudar de pasta. Vou lá para "documents", vou verificar se o meu .py está aqui mesmo. Então coloco o comandinho "dir" aqui. Está aqui: soma_ingredientes. E aí a gente vai executar a soma_ingredientes. Ok. Aí apertado o enter. Bom, gente, deu erro.

00;15;42;25 - 00;16;05;04

Vamos ver aqui, inclusive vamos aprender a ler os erros, tá? Aqui no final já fala pra gente. Deu um erro de nome: name error. Ele fala que soma_ingredientes não foi definido. Aí ele até me fala (muito indiretinho o phyton): você quer dizer "soma_ingrediente". Vamos dar uma olhada lá no nosso código para ver o que aconteceu.

00;16;05;04 - 00;16;30;17

Olha, se a gente observar, a gente chamou a função soma_ingredientes, com o "s" no final: plural. Mas se a gente for olhar aqui, quando a gente definiu, a gente não colocou o s. Então olha a importância de que os nomes têm que ser certinhos, a chamada do nome tem que ser certinha. Então vamos colocar o "s" aqui. Agora tá certinho. Vamos salvar e vamos voltar lá na nossa telinha preta e executar de novo:

00;16;31;14 - 00;16;45;04

phyton soma_ingredientes, enter e aí a gente tem o valor total da compra: 54.5 reais. Será que ficou caro pra comprar os ingredientes para fazer um bolo?