

FOCO NO MERCADO DE TRABALHO

Imprimir

FUNÇÕES EM PYTHON

Vanessa Cadan Scheffer

0

Ver anotações

IMPLEMENTANDO SOLUÇÕES POR MEIO DE FUNÇÕES

Como funções em Python são capazes de lidar com vários tipos de dinâmicas e implementar soluções.



Fonte: Shutterstock.

Deseja ouvir este material?

Áudio disponível no material digital.

DESAFIO

O desenvolvimento de um software pode envolver diversas pessoas e até mesmo diferentes equipes. Orquestrar todos os envolvidos com as etapas do projeto é um desafio em qualquer empresa. Para ajudar nessa missão, a engenharia de software fornece diversas ferramentas, que especificam padrões de projeto e de trabalho. A metodologia ágil é uma dessas ferramentas, que habilita a entrega de partes do software para o cliente. por exemplo. em cada sprint (intervalo de tempo. que

normalmente dura 15 dias) uma nova funcionalidade ou melhoria é entregue. Para conseguir essa agilidade na entrega, pessoas com diferentes expertise (conhecimento) são colocadas para trabalhar juntas, por exemplo, um design trabalha no layout da interface gráfica, um desenvolvedor trabalha nas funcionalidades front-end, enquanto outro trabalha no back-end e podemos ainda ter um DBA (Data Base Administrator) cuidando das conexões com o banco de dados. Dando continuidade ao seu trabalho na empresa de consultora de software, o cliente que fabrica peças automotivas requisitou uma nova funcionalidade para o sistema calcular o total de vendas. Como seus negócios estão expandindo, o cliente solicitou que o sistema seja capaz de receber valores em reais, dólar ou euro, e que seja feita a conversão para o valor em reais. Como essa entrega demanda a construção de uma interface gráfica, além da comunicação com banco de dados, dentre outros requisitos técnicos, você foi alocado em uma equipe ágil para criar a função que fará o cálculo do valor.

o
r
Ver anotações

Após uma primeira reunião, a equipe fez um levantamento de requisitos e concluiu que a função a ser construída precisa considerar os seguintes itens:

- O valor do produto (obrigatório).
- A quantidade do produto (obrigatório).
- A moeda em que deve ser feito o cálculo (obrigatório, sendo padrão o real).
- A porcentagem do desconto que será concedida na compra (opcional).
- A porcentagem de acréscimo, que depende da forma de pagamento (opcional).

Uma conta não pode ter desconto e acréscimo ao mesmo tempo. Nessa primeira versão, você deve considerar o valor do dólar em R\$ 5,00 e do euro 5,70. Ainda não foram definidos os detalhes de como os dados serão capturados e tratados após serem digitados e submetidos. Porém, você deve entregar a versão inicial da função, para que a equipe comece a fazer os primeiros testes.

Qual nome você dará para a função? Como você especificará os parâmetros que a função pode receber? A função retornará um valor ou somente imprimirá na tela o resultado? Hora de colocar a mão na massa e implementar a solução.

RESOLUÇÃO

Em seu novo projeto, você foi designado a implementar uma solução que envolve o cálculo de uma compra. Para esse cálculo existem parâmetros que são obrigatórios e outros opcionais, portanto a função deve ser capaz de lidar com esse tipo de dinâmica. Observe uma possível implementação a seguir.

In [1]:

```
def calcular_valor(valor_prod, qtde, moeda="real", desconto=None, acrescimo=None):
    v_bruto = valor_prod * qtde

    if desconto:
        v_liquido = v_bruto - (v_bruto * (desconto / 100))
    elif acrescimo:
        v_liquido = v_bruto + (v_bruto * (acrescimo / 100))
    else:
        v_liquido = v_bruto

    if moeda == 'real':
        return v_liquido
    elif moeda == 'dolar':
        return v_liquido * 5
    elif moeda == 'euro':
        return v_liquido * 5.7
    else:
        print("Moeda não cadastrada!")
        return 0

valor_a_pagar = calcular_valor(valor_prod=32, qtde=5, desconto=5)
print(f"O valor final da conta é {valor_a_pagar}")
```

```
O valor final da conta é 152.0
```

Sobre a solução proposta, observe o nome da função "calcular_valor", veja que estamos seguindo as recomendações de nomenclatura, usando somente letras em minúsculo e com o underline separando as palavras. Outro detalhe é a utilização do

Ver anotações

verbo no infinitivo "calcular", toda função executa ações, por isso, é uma boa prática escolher verbos infinitos.

A função foi desenvolvida de modo a receber cinco parâmetros, sendo três deles obrigatórios e dois opcionais. Nessa implementação, para construir os parâmetros opcionais atribuímos o valor None às variáveis, nesse caso, como tem um valor padrão, mesmo sendo None, a função pode ser invocada sem a passagem desse parâmetros.

0
Ver anotações

Usamos as estruturas condicionais (if) para verificar se foram passados valores para o desconto ou acréscimo, caso tenha valores, serão diferentes de None e, então, os devidos cálculos são realizados. Após os cálculos de desconto, é feito o teste para saber qual moeda foi usada na compra e fazer a conversão para real.

Muitas vezes, uma solução pode ser implementada de diferentes formas. Observe no código a seguir uma outra implementação, usando o `**kwargs` para os argumentos opcionais. Nesse caso, um dicionário é recebido e precisa ter o valor extraído. Na próxima aula, você aprenderá esse tipo de objeto.

In [2]:

```
def calcular_valor(valor_prod, qtde, moeda="real", **kwargs):
    v_bruto = valor_prod * qtde

    if 'desconto' in kwargs:
        desconto = kwargs['desconto']
        v_liquido = v_bruto - (v_bruto * (desconto / 100))
    elif 'acrescimo' in kwargs:
        acrescimo = kwargs['acrescimo']
        v_liquido = v_bruto + (v_bruto * (acrescimo / 100))
    else:
        v_liquido = v_bruto

    if moeda == 'real':
        return v_liquido
    elif moeda == 'dolar':
        return v_liquido * 5
    elif moeda == 'euro':
        return v_liquido * 5.7
    else:
        print("Moeda não cadastrada!")
        return 0

valor_a_pagar = calcular_valor(valor_prod=32, qtde=5, desconto=5)
print(f"O valor final da conta é {valor_a_pagar}")
```

O valor final da conta é 152.0

0
Ver anotações

DESAFIO DA INTERNET

Ganhar habilidade em programação exige estudo e treino (muito treino). Acesse a biblioteca virtual no endereço <http://biblioteca-virtual.com/> e busque pelo livro:

MANZANO, J. A. N. G; OLIVEIRA, J. F. de. **Estudo Dirigido de Algoritmos**. 15. ed. São Paulo: Érica, 2012. Na página 209, do capítulo 12 da referida obra: **Sub-rotinas do tipo função**. você encontrará o exemplo 3. que te desafia a criar uma função

Python3

main.py - inflammation_01.csv

3

CodeRun▼Share▼

Remix+

0

<>

Escolher arquivo

Nenhum arquivo escolhido

Ver anotações