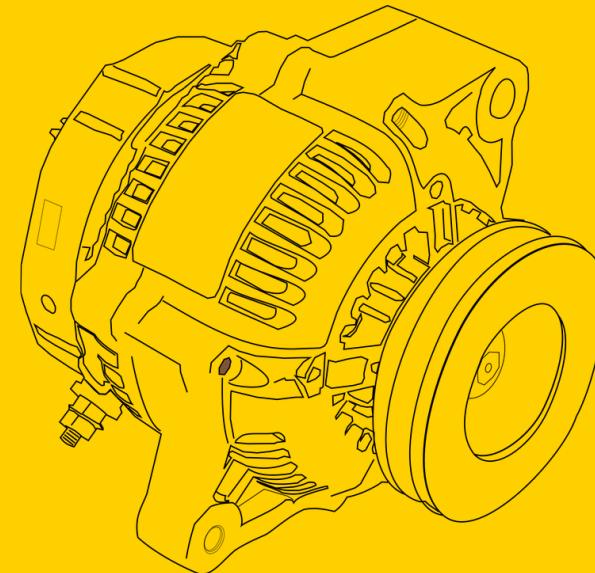




»»» **Funciona!**

Módulo 01- Aula 12 - HARD



Todos os direitos reservados
©2023 Resilia Educação



Qual o meu objetivo de hoje?

HARD

- Entender como as funções tornam os códigos legíveis, reutilizáveis e testáveis;
- Compreender os principais conceitos de funções: nome, declaração, chamada, parâmetros e retorno;
- Implementar funções em situações específicas solicitadas.

SOFT

- Defender a necessidade de feedback para o trabalho em equipe no contexto do mundo corporativo;
- Listar os diferentes tipos de feedback;
- Categorizar experiências anteriores de feedbacks recebidos, de acordo com princípios aprendidos;
- Experimentar a prática de feedback para trabalho em equipe

Nossos códigos estão ruins!



Problema dos nossos códigos

Nossos códigos estão muito **desorganizados** e **soltos**!

Precisamos de uma forma de criar **pequenos componentes** de código que sejam:

- **Legíveis**: Ao bater o olho já sabemos o que o código faz
- **Reutilizáveis**: Poder transportar de um programa para outro sem dificuldade
- **Testáveis**: Saber exatamente qual é o resultado esperado como resposta

Funções!



Funções: definição

Funções são **pequenos blocos de código** que executam **uma tarefa** e possuem um **nome**.

As **vantagens** de utilizar funções são:

- **Legibilidade de código**
- **Reutilização de código**
- **Testabilidade de código**



Funções: declaração

Para **declarar** uma **função** em Python utilizamos a palavra reservada **function** seguida do **nome da função, parênteses** (com **parâmetros** necessários dentro **separados por vírgulas**) e **chaves** envolvendo o **bloco de código** da função. Uma função também pode **retornar** um **valor**.

Ex:

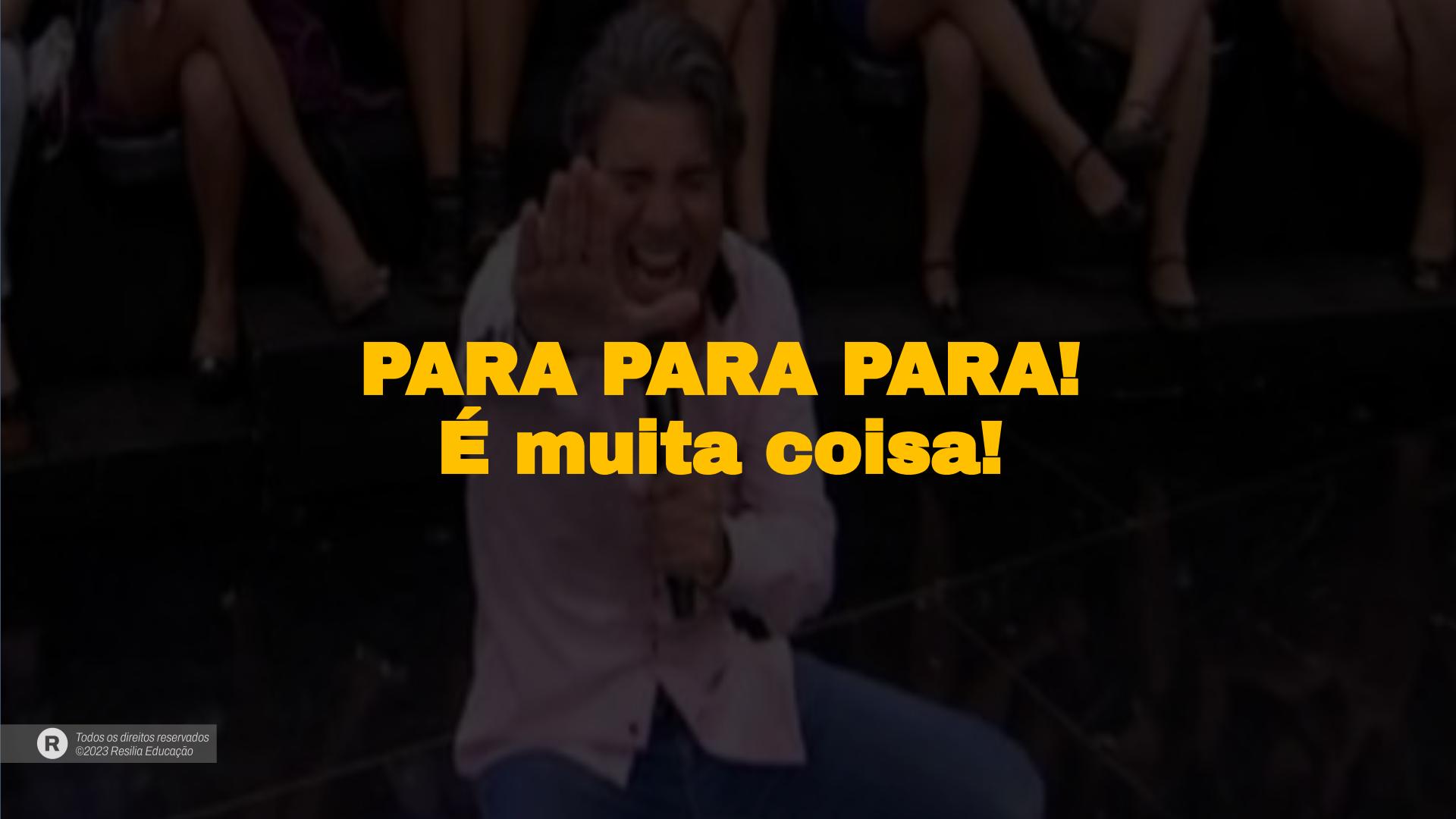
```
def criaTextoPreco(item, preco):  
    return ('O item ' + item + ' custa R$' + preco)
```

Funções: invocação

Para invocar/**chamar** uma **função**, utilizamos o **nome dela** e passamos **argumentos** entre **parênteses**.

Ex:

```
//Retornará: 'O item Tomate custa R$0.98'  
criaTextoPreco('Tomate', 0.98)
```



PARA PARA PARA!
É muita coisa!





Vamos por partes

Parâmetros

Parâmetros

Os **parâmetros** são as **variáveis curingas** nas funções. A **declaração** deles é realizada **após o nome** da função, entre **parênteses** e com **separação** por **vírgulas**. No bloco de código, eles **representam** o **valor** utilizado quando a função é **chamada**.

Ex:

```
def criaTextoPreco(item, preco):  
    return ('O item ' + item + ' custa R$' + preco)
```

Retorno



Retorno

O **retorno** de uma função é o **valor** que ela **retornará** após sua chamada e **execução** do seu bloco de código. A palavra reservada **return** é utilizada para marcar o **valor/variável** que será retornado. Podemos ter **mais de um return** em uma função.

Ex:

```
def criaTextoPreco(item, preco) :  
    return ('O item ' + item + ' custa R$' + preco)
```



Argumentos

Argumentos

Os **argumentos** são os **valores** utilizados quando **chamamos** uma **função**. Eles **substituem** os **parâmetros** no bloco de **código executado**.

Ex:

```
//Retornará: 'O item Tomate custa R$0.98'  
criaTextoPreco('Tomate', 0.98)
```



Funções que conhecemos

Nós **já utilizamos/chamamos** algumas **funções** nas aulas anteriores!

Ex:

```
print("Olá mundo!")  
saborPizza = input("Qual sabor de pizza você quer?")
```

Melhoria: sem função

O que o código abaixo faz?

```
conceito = ''  
nota = input()  
if (nota >= 9):  
    conceito = 'A'  
elif (nota >= 8):  
    conceito = 'B'  
elif (nota >= 6):  
    conceito = 'C'  
elif (nota >= 4):  
    conceito = 'D'  
else:  
    conceito = 'E'  
print(conceito)
```



Melhoria: com função

Sabendo que a **função conceitoDeNota** foi **declarada** e que ela **recebe** uma **nota** e **retorna** o **conceito** associado a ela, O que o código abaixo faz?

Ex:

```
nota = input()  
print(conceitoDeNota(nota))
```



O liquidificador



Liquidificador

A função do liquidificador é **liquidificar**. Ele tritura, transforma e liquidifica tudo o que colocamos dentro dele. Então, **se colocarmos banana, leite e achocolatado** dentro dele teremos uma **vitamina de banana**.

Ex:

```
def liquidificar(item1, item2, item3):
    if (item1=='Banana' and item2=='Leite' and item3=='Achocolatado'):
        return ('Vitamina de banana')
    else:
        return ('Liquidifiquei tudo!')
```



Funções - resumo

Funções são **caixas mágicas**! Elas **recebem** algo na **entrada** e **transformam** na **saída**.



Atividade: Funcionando

→ **O QUE FAZER?**

Implementar as funções que estão sendo pedidas, utilizando os parâmetros e retornos definidos e testá-las!;

→ **COMO FAZER?**

Em um arquivo .py:

- Crie uma função chamada dirigirOuBeber para verificar se a pessoa já pode dirigir e beber. Sua função deve:
 - Receber uma idade
 - Caso a idade fornecida seja maior que 18, retorne “Já pode dirigir ou beber”
 - Caso contrário, retorne “Não pode nem dirigir nem beber”



Atividade: Funcionando

→ **COMO FAZER?**

Em um arquivo .py:

- Crie uma outra função chamada previsaoRodagem para verificar o quanto será possível rodar sem abastecer. Sua função deve:
 - Receber a quantidade de litros de gasolina no tanque do carro
 - Receber a quilometragem média por litro
 - Retornar a estimativa de km que podem ser rodados nas circunstâncias fornecidas

→ **FECHAMENTO**

Compartilhar, com a turma, as dúvidas e as soluções encontradas.



Funções



Parâmetros

Argumentos



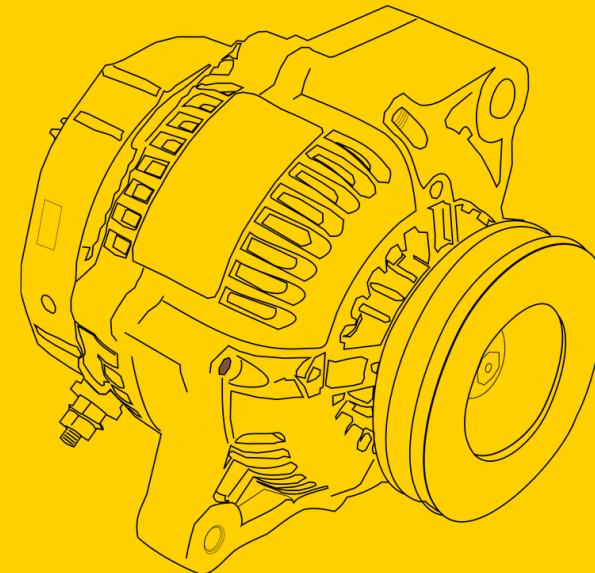
Retorno





»»» **Funciona!**

Módulo 01- Aula 12 - SOFT



Todos os direitos reservados
©2023 Resilia Educação



MUNDO DOS NEGÓCIOS



FEEDBACK

“Realimentação”: qualquer informação que você recebe sobre si mesmo.



Os tipos de feedback!



Reconhecimento

Reconhecer, motivar, agradecer



Orientação

Ajudar o receptor a aumentar seus conhecimentos, aprimorar suas técnicas, aperfeiçoar suas capacidades. Ou aborda os sentimentos do receptor, ou ainda um desequilíbrio no relacionamento.



Avaliação

Classificar ou comparar com um padrão, ajustar expectativas, subsidiar a tomada de decisões

Princípios do Feedback



Atividade: Princípios do feedback

→ **O QUE FAZER?**

Refletir sobre diferentes tipos de feedback

→ **COMO FAZER?**

→ Em grupos, compartilhar experiências e discutir exemplos de quando receberam feedback eficaz/útil e feedback ineficaz/prejudicial, ou seja, que causou incômodo

→ **FECHAMENTO**

Compartilhar as ideias do grupo com a turma, para preenchimento, pela facilitação, do slide síntese.

Atividade: Princípios do feedback

Eficaz	Ineficaz
•	•

CHARACTER SELECT

Escolha sua dupla!

Seu parceiro será responsável por acompanhar seu desenvolvimento na jornada e vice-versa.



Todos os direitos reservados
©2023 Resilia Educação



Confirm



Back

Papéis e responsabilidades

Escolha uma dupla e marque um encontro semanal para as sessões de mentoria.
Nesses encontros, cada um deverá desempenhar dois papéis: o de “mentorado” e de “mentor”.

Quem recebe mentoria:

- Compartilhar com a dupla o plano de aprendizagem, evolução e dificuldades na semana;
- Combinar planos de ação e realizar dentro do tempo estipulado.
- Pedir ajuda para realizar algum plano, caso necessário.



Quem mentora:

- Acompanhar a realização de cada plano de ação;
- Fazer anotações de percepções sobre o desempenho da dupla;
- Oferecer ajuda, direcionamento, feedbacks e novos pontos de vista para realização de um plano de ação.



Até a próxima e
#confianoprocess

