Aula de Python

Python básico: Review estruturas + funções

WOMAKERS CODE

Tipos básicos

- **String (f'string)**
- **Int**
- **Float**
- **Bool ou Boolean**

Ex:

```
#Concatenar
print('O faturamento da loja foi ' + '1000')

#Verificar se um texto está dentro do outro
print('@' in 'lira@gmail.com')
```

Listas - Estruturas de dados

lista = [valor, valor, valor, valor, ...]

- Lista é um dos objetos mais importantes de Python!

- Quando importamos uma base de dados para o Python, normalmente ele é lido como uma "lista" ou como alguma "variação de lista".

- Listas em Python foram feitas para serem homogêneas, apesar de aceitarem valores heterogêneos.

Descobrindo o índice dentro de uma lista.
Métodos: lista.append(x), lista.remove

Tuplas – Estruturas de dados

tupla = (valor, valor, valor, ...)

Diferença

Parece uma lista, mas é imutável.

Vantagens:

- Mais eficiente (em termos de performance).
- Protege a base de dados (por ser imutável).
- Muito usado para dados heterogêneos.

Dicionários - Estruturas de dados

Estrutura:

```
dicionario = {chave1: valor1, chave2: valor2, chave3: valor3, chave4: valor4 ...}
```

Vantagens e Desvantagens

- Não devem ser usados para pegar itens em uma determinada ordem.
- Podem ter valores heterogêneos (vários tipos de valores dentro de um mesmo dicionário: inteiros, strings, listas, etc).
- Chaves são únicas obrigatoriamente.
- Mais intuitivos de trabalhar.

Condicionais

```
if condição:
    o que fazer caso a condição seja verdadeira.
```

Digamos que você trabalha na Amazon (que tem centenas de milhares, se não milhões de produtos) e está analisando o resultado de vendas dos produtos.
Você precisa criar um programa que vai analisar o resultado de vendas dos produtos da Amazon em um mês.
Para simplificar vamos pensar em um único produto: um Iphone.

Meta de Vendas do Iphone = 50.000 unidades.

Quantidade vendida no Mês = 65.300 unidades

O seu programa deve avisar (usaremos o print) caso o produto tenha batido a meta do mês. Então devemos fazer:

- Caso o produto tenha batido a meta, devemos exibir a mensagem: "Batemos a meta de vendas de Iphone, vendemos {} unidades"
- Se ele não bateu a meta do mês, o seu programa não deve retornar "Não batemos a meta".

Apenas UMA função!

Funções - é uma ação (passiva ou ativa)!

As functions são blocos de código que servem 1 único propósito, fazem uma ação específica.

```
def nome_funcao():
    faça alguma coisa
    faça outra coisa
    return valor_final
```

camelCase
snake_case
PascalCase

Escreva o que você quer que ela faça.

Ex da checagem de e-mail!

Funções - Retornar um valor.

```
def nome_funcao():
    return valor_final
```

Exemplo:

- Exemplo: vamos criar uma função de cadastro de um Produto. Essa função deve garantir que o produto cadastrado está em letra minúscula.

```
def cadastrar_produto():
   produto = input('Digite o nome do produto que deseja cadastrar')
   produto = produto.casefold()
   produto = produto.strip()
   return produto
```

Funções - Parâmetros e argumentos

```
def minha_funcao(parametro1, parametro2, parametro3):
    return parametro1 + parametro2 + parametro3
```

Exemplo da função print().

Vamos criar uma function com Parâmetro?

Digamos que estamos criando um programa para categorizar os produtos de uma revendedora de bebidas.

Cada produto tem um código. O tipo de produto é dado pelas 3 primeiras letras do código.

Ex:
Vinho -> BEB12302
Cerveja -> BEB12043
Vodka -> BEB34501

Guaraná -> BSA11104
Coca -> BSA54301
Sprite -> BSA34012
Água -> BSA09871

Repare que bebidas não alcóolicas começam com BSA e bebidas alcoólicas começam com BEB.

Crie um programa que analise uma lista de produtos e envie instruções para a equipe de estoque dizendo quais produtos devem ser enviados para a área de bebidas alcóolicas.

Quantidades indefinidas de Argumentos

Quando você quer permitir uma quantidade indefinida de argumentos, usa o * para isso.

```
*args para positional arguments -> argumentos vêm em formato de tupla

def minha_funcao(*args):
    ...


**kwargs para keyword arguments -> argumentos vêm em formato de dicionário

def minha_funcao(**kwargs):
```

Isso é tudo pessoal!

Qualquer um consegue escrever códigos que um computador entenda, mas poucos fazem códigos que outro ser humano entenda.

Uncle BOB.