

Data Science – Princípios e Técnicas

Agosto
2024



Data Science



Onde me encontrar:

<https://www.linkedin.com/in/marco-mialaret-junior/>

e

<https://github.com/MatmJr>

Data Science

Listas

São definidas com colchetes `[]` e seus elementos podem ser alterados, ou seja, você pode adicionar, remover ou alterar itens da lista depois de ela ser criada.

Data Science

append(x): Adiciona um item ao final da lista.

extend(iterable): Estende a lista adicionando todos os itens do iterável fornecido ao final.

insert(i, x): Insere um item em uma posição específica.

Data Science

remove(x): Remove o primeiro item da lista cujo valor é igual a x.

pop([i]): Remove o item na posição dada e o retorna. Se nenhum índice for especificado, remove e retorna o último item da lista.

Data Science

```
frutas = ["maçã", "banana", "laranja"]
```

```
frutas.append("uva")
```

```
frutas.extend(["manga", "abacaxi"])
```

```
frutas.insert(1, "morango")
```

```
frutas.remove("laranja")
```

```
fruta_removida = frutas.pop(2)
```

Data Science

Tuplas

São definidas com parênteses `()` e, uma vez criadas, não é possível modificar seus elementos. Isso significa que você não pode adicionar, remover ou alterar itens da tupla depois de ela ser definida.

Data Science

Tuplas são usadas quando você quer que os dados permaneçam constantes e não sejam modificados.

Data Science

```
livro = ("O mundo de Sofia", "Jostein Gaarder ", 1991)
```

```
titulo = livro[0]
```

```
autor = livro[1]
```

```
ano = livro[2]
```

```
print(f"Título: {titulo}")
```

```
print(f"Autor: {autor}")
```

```
print(f"Ano de publicação: {ano}")
```

Data Science

Tentativa de modificar a tupla resultará em erro

```
livro[2] = 1950
```

Obs.: Isso causará um erro, pois tuplas são imutáveis

Data Science

Condicionais

A estrutura condicional if-elif-else em Python é usada para executar diferentes blocos de código com base em uma ou mais condições. É uma das principais maneiras de controlar o fluxo de execução do programa, permitindo que ele tome decisões.

```
if CONDIÇÃO :  
    BLOCO DE CÓDIGO  
elif CONDIÇÃO :  
    BLOCO DE CÓDIGO  
else :  
    BLOCO DE CÓDIGO
```

Data Science



```
idade = int(input("Digite sua idade: "))
```

```
if idade < 18:  
    print("Você é menor de idade.")  
elif idade == 18:  
    print("Você tem 18 anos.")  
else:  
    print("Você é maior de idade.")
```

Laço de repetição for

Um laço for no Python é uma estrutura de controle que permite iterar sobre uma sequência de elementos, executando um bloco de código repetidamente para cada elemento na sequência.

Data Science

Essa sequência pode ser uma lista, uma tupla, um dicionário, um conjunto ou qualquer objeto iterável. O laço for é usado para realizar uma operação, ou uma série de operações, em cada item da sequência, uma vez por vez.

Data Science



for **variável** in **sequência**:

variável: É um nome que você dá para representar o elemento atual sendo processado na sequência durante as iterações do laço.

sequência: É a sequência de elementos que você deseja iterar. Pode ser qualquer objeto iterável do Python.

Data Science

```
nomes = ["Ana", "Bruno", "Carlos"]  
for nome in nomes:  
    print(nome)
```


Data Science

Criando Listas no python:

```
lista = []
```

```
for item in range(10):  
    lista.append(item**2)
```

Data Science

Compreensão de listas:

List comprehension é uma técnica em Python que permite criar novas listas aplicando uma expressão a cada item de uma sequência ou iterável, de forma concisa e legível. A sintaxe básica é:

Data Science

[nova_expressão for item in iterável if condição]

Essa estrutura permite transformar e filtrar elementos de maneira compacta, unificando o processo de criação da lista em uma única linha de código.

Data Science

`[item**2 for item in range(10) if item%2]`

Data Science

O laço de repetição while

No Python tem basicamente a mesma estrutura do for, porém, como no for ou if, não se utilizam os parênteses para definir a condição e seu bloco de instrução também é definido com a indentação.

Data Science

```
i = 0
while i < 10:
    print ("i = ",i)
    i += 1
```

Funções

Assim como em outras linguagens, uma função pode receber alguns parâmetros e devolver um resultado. A sintaxe inicia-se com a palavra `def`, seguida do nome da função e dos parênteses que podem conter seus parâmetros.

Data Science

A definição de uma função começa com a palavra-chave `def`, seguida pelo nome da função e por parênteses que podem conter os parâmetros. O corpo da função é então indentado e contém as instruções que serão executadas quando a função for chamada.

Data Science

```
def saudacao(nome):  
    return f"Olá, {nome}!"
```

Dúvidas?



81 98160 7018

marcomialaret@gmail.com

