EXPLORANDO PYTHON

MANUAL PRÁTICO PARA INICIANTES

DANIELLY CRUZ MANOEL

Python para iniciantes

Capítulo 1: Introdução ao Python

Capítulo 2: Variável e Tipos de Dados

Capítulo 3: Estruturas de Controle

Capítulo 4: Funções e Módulos

Capítulo 5: Listas e Dicionários

01

INTRODUÇÃO AO PYTHON

Para começar a programar em Python, você precisa ter o interpretador Python instalado em seu sistema. Você pode baixar e instalar a versão mais recente do Python em python.org.Ambiente de Desenvolvimento.

Um ambiente de desenvolvimento integrado (IDE) facilita a escrita de código. Recomendações populares incluem PyCharm, VS Code, e Jupyter Notebook.Hello, World!

Vamos começar com o famoso programa "Hello, World!" em Python:

```python
print("Hello, World!")
```

02

VARIÁVEIS E TIPOS DE DADOS

Variáveis

As variáveis são usadas para o armazenamento de dados. Exemplo:

```
idade = 25
nome = "João"
```

Tipos de Dados

Os diferentes tipos de dados, inclui inteiro, flutuantes, strings, listas, tuplas, dicionários, entre outros. Exemplos:

```python
inteiro = 10
flutuante = 3.14
texto = "Olá, mundo!"
 lista = [1, 2, 3, 4]
```

03

LISTAS E DICIONÁRIOS

Listas

São usadas para armazenar múltiplos itens em únicas variável. Elas são ordenadas, mutáveis (podem ser alteradas) e permitem itens duplicados.

```python
# Criando uma lista de frutas
frutas = ["maçã", "banana", "laranja", "uva"]

# Acessando um item da lista
print(frutas[1])  # Saída: banana

# Adicionando um item à lista
frutas.append("manga")
print(frutas)  # Saída: ['maçã', 'banana', 'laranja', 'uva',
'manga']

# Removendo um item da lista
frutas.remove("laranja")
print(frutas)  # Saída: ['maçã', 'banana', 'uva', 'manga']

# Iterando sobre a lista
for fruta in frutas:
    print(fruta)
```

Explicação

1- Criamos uma lista chamda 'frutas' com quatro elementos.
2- Usamos 'print(frutas[1])' para acessar e imprimir o segundo item da lista (índices começam em 0).
3- Usamos 'frutas.append("manga')' para adicionar "manga" ao finaç de lista.
4- Usamos 'frutas.remove("laranja")' para remover "laranja" da lista.
5- Iteramos sobre a lista com um laço 'for', imprimindo cada fruta.

Dicionários

São usados para armazenar dados em pares chave-valor.

```python
# Criando um dicionário
pessoa = {
    "nome": "Ana",
    "idade": 28,
    "cidade": "São Paulo"
}

# Acessando um valor pelo chave
print(pessoa["nome"])  # Saída: Ana

# Adicionando um novo par chave-valor
pessoa["profissão"] = "Engenheira"
print(pessoa)  # Saída: {'nome': 'Ana', 'idade': 28, 'cidade':
'São Paulo', 'profissão': 'Engenheira'}

# Removendo um par chave-valor
del pessoa["idade"]
print(pessoa)  # Saída: {'nome': 'Ana', 'cidade': 'São Paulo',
'profissão': 'Engenheira'}

# Iterando sobre o dicionário
for chave, valor in pessoa.items():
    print(f"{chave}: {valor}")
```

Explicação

1- Criamos um dicionário chamado 'pessoa' com três pares chave-valor.
2- Usamos 'print(pessoa["nome"])' para acessar e imprimir o valor associado à chave "nome".
3- Adicionamos um novo par chave-valor ao dicionário com 'pessoa{"profissão"] = "Engenheira".
4- Removemos o par chave-valor com chave "idade" usando 'del pessoa["idade"]'.
5- Iteramos sobre o dicionário com um laço 'for', imprimindo cada chave e valor.

04

ESTRUTURAS DE CONTROLE

CONDICIONAL IF E ELSE

O if e o else são usados para tomar decisões com base em um condições, sendo 'if' o 'se' e o 'else' o 'se não'.

```
idade = 18
if idade >= 18:
  print('você é maior de idade.')
else:
  print('Você é menor de idade.')
```

No exemplo tem um variável do tipo inteiro, chamada idade = 18.

E ser(if) for maior ou igual a 18, terar uma mensagem. Ser for menor outro mensagem.

Laços de Repetição

Laço For

Geralmente usado quando você precisa iterar sobre uma sequência de elementos, como um lista, tupla ou string. Você sabe de antemão quantas vezes o bloco de código será executado.

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

for fruta in frutas:
    print(fruta)
```

Explicação:

1- Definimos um lista chamada 'frutas' com quatro elementos.
2- Usamos um laço 'for' para iterar sobre cada elemento da lista.
3- Para cada iteração, a variável 'fruta' assume o valor do elemento atual da lista, e o comando 'print(fruta)' imprime esse valor.

```
maçã
banana
laranja
uva
```

Laço While

Mais adequado quando você não sabe extamente quantas vezes o bloco de código deve ser executado.

```python
contador = 1

while contador <= 5:
    print(contador)
    contador += 1
```

Explicação:

1- Inicializamos a variável 'contador' com o valor 1.
2- O laço, 'while' continua executando enquanto a condição 'contador <= 5' for verdadeira.
3- Dentro do laço, usamos 'print(contador)' para imprimiro valor atual de 'contador'.
4- Incrementamos 'contador' em 1 a cada iteração 'contador +=1'.

05

FUNÇÕES E MÓDULOS

Funções

São blocos de código que realizam uma tarefa específica e podem ser reutilizados em deferentes partes de um programa. Definir fuções torna o código mais modular e fácil de entender.

```python
def soma(a, b):
    resultado = a + b
    return resultado

# Usando a função
numero1 = 5
numero2 = 3
print(soma(numero1, numero2))
```

Explicação:

1- Definimos a função 'soma' que recebe dois parâmetro, 'a' e 'b'.
2- Dentro da função, somamos 'a' e 'b' e armazenamoso resultado na variável 'resultado'.
3- A função retrona o valor de 'resultrado'.
4- For a da função, chamamos 'soma' com os argumentos 'numero1' e 'numero2' e imprimimoso resultado.

```
8
```