

PYTHON PARA INICIANTEs

Siga os Passos de Nadine



Márcio Cavalcante



01

INTRODUÇÃO

INTRODUÇÃO

Se você está começando a sua jornada no mundo da programação, o Python é uma excelente escolha. Ele é simples, poderoso e muito usado no mercado.

Neste ebook, vamos explorar os conceitos básicos da linguagem de maneira direta e prática, para que você comece a programar de forma eficiente e divertida.

02

VARIÁVEIS

Guardando Dados no Python

VARIÁVEIS

Guardando Dados no Python

Em Python, as variáveis são como caixinhas onde você pode armazenar dados. Elas podem guardar números, textos ou outros tipos de informações. Para criar uma variável, você apenas precisa dar um nome a ela e atribuir um valor.

Variáveis

```
nome = "Nadine"
idade = 5
print(nome) # Saída: Nadine
print(idade) # Saída: 5
```

No exemplo acima, criamos duas variáveis: **nome** e **idade**. A variável **nome** guarda um texto (string), e a variável **idade** guarda um número (inteiro).

03

OPERADORES

Fazendo Cálculos e Comparações

OPERADORES

Fazendo Cálculos e Comparações

Os operadores em Python permitem que você faça cálculos matemáticos, combine textos e faça comparações. Vamos ver os principais tipos de operadores.

Operadores Aritméticos

São usados para realizar cálculos. Como somar, subtrair, multiplicar, dividir, entre outros.

```
Operadores

a = 10
b = 5
soma = a + b # Soma
multiplicacao = a * b # Multiplicação
divisao = a / b # Divisão
print(soma, multiplicacao, divisao)
```

OPERADORES

Fazendo Cálculos e Comparações

Operadores de Comparação

São usados para comparar valores e retornar se são iguais, maiores, menores, etc.

```
Operadores

x = 10
y = 20
print(x == y)  # False (igualdade)
print(x < y)   # True (menor que)
print(x > y)   # False (maior que)
```


04

CONDICIONAIS

Tomando Decisões no Código

CONDICIONAIS

Tomando Decisões no Código

Condicionais permitem que se faça decisões no código. Com o **if**, você pode executar um bloco de código se uma condição for verdadeira, e com **else**, pode executar algo diferente caso a condição seja falsa.

```
idade = 15

if idade >= 18:
    print("Você é maior de idade!")
else:
    print("Você é menor de idade!")
```

Aqui, o código verifica se a pessoa tem 18 anos ou mais. Se sim, imprime "Você é maior de idade!". Caso contrário, "Você é menor de idade!".

05

LAÇOS DE REPETIÇÃO

Automatizando Tarefas com o *for* e *while*

LAÇOS DE REPETIÇÃO

Automatizando Tarefas com o *for* e *while*

Quando você precisa repetir uma ação várias vezes, pode usar os laços de repetição. O Python oferece dois tipos principais: ***for*** e ***while***.

Laço *for*:

Ideal quando você sabe quantas vezes precisa repetir algo. Por exemplo, se quiser imprimir uma lista de números.

Laços de Repetição

```
for i in range(5): # repete 5 vezes, de 0 até 4
    print(i)
```

O `range(5)` cria uma sequência de números de 0 até 4. O laço ***for*** percorre cada número e imprime ele.

LAÇOS DE REPETIÇÃO

Automatizando Tarefas com o *for* e *while*

Laço *while*:

Usado quando você não sabe exatamente quantas vezes precisa repetir, mas quer que a repetição continue enquanto uma condição for verdadeira.

Laços de Repetição

```
contador = 0
while contador < 5:
    print(contador)
    contador += 1 # Incrementa o contador de 1 em 1
```

Aqui, o laço ***while*** vai continuar enquanto contador for menor que 5. A cada iteração, o valor de contador aumenta em 1.

06

COMENTÁRIOS

Tornando o Código Legível

COMENTÁRIOS

Tornando o Código Legível

Comentários são textos colocados no código para explicar o que está acontecendo. Eles não afetam a execução do programa, mas ajudam a entender o que o código faz.

Sempre use comentários para descrever o que está fazendo. Isso facilita a leitura e manutenção do código.

Comentários

```
# Este é um comentário. Ele não afeta o código.  
  
# Vamos somar dois números  
a = 10 # Primeiro número  
b = 5  # Segundo número  
resultado = a + b # Somamos a e b  
print(resultado)
```

07

CONCLUSÃO

CONCLUSÃO

Conteúdo criado por IA

Agora que você já conhece alguns dos principais conceitos do Python, pode começar a criar seus próprios programas. Experimente modificar os exemplos que mostramos e crie algo novo. Lembre-se de que a prática é essencial para dominar a programação, então, continue explorando!

Com esses conhecimentos básicos, você está pronto para explorar mais recursos avançados da linguagem Python e transformar suas ideias em código. Boa jornada no mundo da programação!

Obs.: Conteúdo criado apenas para fins didáticos.