

# PYDEX



O Guia Completo de para Treinadores  
Python

**ARTHUR NEVES**

# PRINCIPAIS COMANDOS

Este eBook serve como uma introdução aos comandos em Python, estruturado para cobrir desde os conceitos básicos até aplicações mais avançadas. Ao longo deste guia, você encontrará explicações claras e exemplos práticos de código para facilitar o aprendizado e a aplicação imediata do Python em seus projetos.





# Fundamentos



01

# Variáveis

## Declarando Variáveis

Para declarar variáveis em Python, basta atribuir um valor a um nome. Python é uma linguagem dinamicamente tipada, então você não precisa especificar o tipo de dados.

```
variables.py

x = 10 # Inteiro
y = 20.5 # Ponto flutuante
```



# Variáveis

## Atribuindo Valores

Você pode reatribuir valores para uma variável já existente a qualquer momento.



variables.py

```
x = 15 # x agora é 15
```



# Tipos de dados

## Integers

São números inteiros, positivos ou negativos, sem parte decimal.

```
data_types.py  
  
age = 25
```



# Tipos de dados

## Floats

Representam números reais e podem incluir uma parte decimal.

 data\_types.py

```
height = 175.5
```



# Tipos de dados

## Strings

Texto entre aspas simples ou duplas.

```
data_types.py  
  
name = "John Doe"
```





# Tipos de dados

## Booleans

Valores lógicos que podem ser True ou False.



data\_types.py

```
is_online = True  
is_online = False
```



# Operadores

## Operadores aritméticos

Usados para realizar operações matemáticas como adição, subtração, multiplicação e divisão.

```
operators.py

sum = 10 + 5 # 15
product = 10 * 5 # 50
```



# Operadores

## Operadores de comparação

Usados para comparar valores. Retorna True ou False.



operators.py

```
print(10 > 5) # True  
print(10 = 5) # False
```



# Operadores

## Operadores lógicos

Usados para combinar condições lógicas.



operators.py

```
print(10 > 5 and 10 < 20) # True  
print(10 < 5 or 10 < 20) # True
```



# Controle de fluxo

## Declarações if-else

Permite que você execute condicionalmente blocos de código.

control\_flow.py

```
if age ≥ 18:  
    print("Adult")  
else:  
    print("Minor")
```





# Controle de fluxo

## for Loops

Executa um bloco de código várias vezes, útil para iterar sobre uma sequência.



control\_flow.py

```
for i in range(5):  
    print(i)
```



# Controle de fluxo

## while Loops

Repete um bloco de código enquanto uma condição é verdadeira.

```
control_flow.py

i = 0
while i < 5:
    print(i)
    i += 1
```



# Funções



02

# Definindo Funções

Você pode definir funções para encapsular código para reutilização.

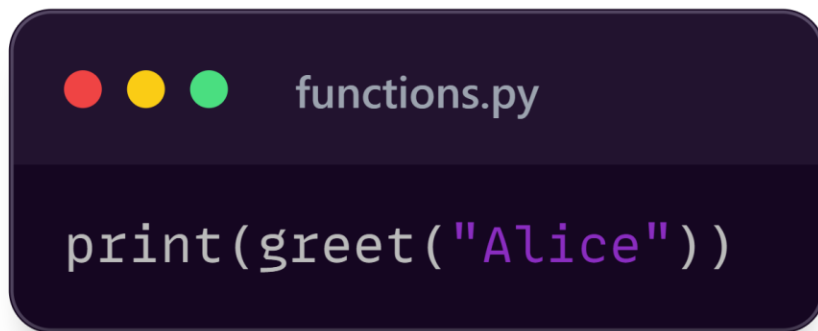
```
functions.py

def greet(name):
    return "Hello, " + name
```



# Chamando Funções

Chama uma função para executar seu código.



```
functions.py  
  
print(greet("Alice"))
```





# Parâmetros de Função

Passa informações para funções através de parâmetros.



functions.py

```
def greet(name, message):  
    return f"{message}, {name}"
```



# Atribuindo Valores

Retorna um valor de uma função.



```
functions.py
```

```
def sum(a, b):  
    return a + b
```



# AGRADECIMENTOS



# Obrigado por ler até aqui

Esse Ebook foi gerado por IA, e diagramado por humano.

Esse conteúdo foi gerado com fins didáticos de construção, não foi realizada uma validação cuidadosa humana no conteúdo e pode conter erros gerados por uma IA.

Os próximos capítulos serão feitos no GitHub e o material será escrito em markdown e teremos os exemplos de código no Google Colab para que todos tenham acesso mais fácil

<https://github.com/ArthNeves/ebooks/blob/master/python/pydex.md>

