

# Python

## Básico





## Introdução

Variáveis e Tipos Embutidos

1 Tipos Embutidos (built-ins)

2 Variáveis

3 Para Saber Mais: Nomes de Variáveis

4 Instruções

5 Operadores Aritméticos

6 Strings

7 Entrada do Usuário

8 Constantes

9 Comando if

10 Convertendo uma String para Inteiro

11 O Comando elif

12 Exercícios - Jogo da Adivinhação

13 Comando while

14 Exercícios - Jogo com while

15 Comando for

16 Exercícios - Utilizando o for no jogo

## Conclusão

# Olá!

Bem-vindo ao nosso guia completo de Python para iniciantes! Este eBook foi criado para ajudá-lo a entender os fundamentos de Python, uma das linguagens de programação mais populares e versáteis do mundo. Cada tópico inclui explicações detalhadas e exemplos de código para facilitar o aprendizado. Vamos começar!

## Requisitos

### 1. Computador

- Um computador com qualquer um dos sistemas operacionais: Windows, macOS ou Linux.

### 2. Editor de Código

- Um editor de código para escrever e executar seus scripts Python. Recomendações:
  - Visual Studio Code
  - PyCharm
  - Sublime Text

### 3. Python Instalado

- Python 3.x deve estar instalado e configurado no seu sistema. Você pode baixar o Python no site oficial.

### 4. Conhecimento Básico de Programação

- Noções básicas sobre o que são variáveis, loops e condições podem ajudar, mas não são estritamente necessárias, já que este eBook cobre esses conceitos.

### 5. Disposição para Aprender

- Curiosidade e paciência são essenciais para seguir em frente, praticar os exemplos e resolver os exercícios.

Seguindo este diagrama e preparando o ambiente de desenvolvimento conforme os requisitos, você estará pronto para aproveitar ao máximo o conteúdo deste eBook. Boa leitura e bons estudos!

# O que é Python?

Python é uma linguagem de programação de alto nível, interpretada e de propósito geral. Criada por Guido van Rossum e lançada em 1991, Python enfatiza a legibilidade do código, utilizando uma sintaxe que permite aos programadores expressarem conceitos em menos linhas de código do que em outras linguagens, como C++ ou Java. É uma das linguagens mais populares do mundo, utilizada em diversas áreas como desenvolvimento web, análise de dados, inteligência artificial e automação.



Guido van Rossum



Python Logo

# Variáveis e Tipos Embutidos

## 1 Tipos Embutidos (*built-ins*)

Python possui vários tipos de dados embutidos, que são usados para armazenar valores diferentes. Aqui estão alguns dos principais tipos:

Exemplos de Tipos Embutidos:

```
1 # Tipo inteiro
2 numero_inteiro = 10
3 print(type(numero_inteiro)) # <class 'int'>
4
5 # Tipo float
6 numero_float = 10.5
7 print(type(numero_float)) # <class 'float'>
8
9 # Tipo string
10 texto = "Olá, Mundo!"
11 print(type(texto)) # <class 'str'>
12
13 # Tipo booleano
14 verdadeiro = True
15 print(type(verdadeiro)) # <class 'bool'>
16
```

# Variáveis e Tipos Embutidos

## 2 Variáveis

Variáveis são usadas para armazenar informações que podem ser referenciadas e manipuladas em um programa. Em Python, a atribuição de variáveis é feita usando o operador “=”.

*Exemplo de Declaração de Variáveis:*

```
1 # Declarando variáveis
2 nome = "Alice"
3 idade = 30
4 altura = 1.65
5
6 # Imprimindo as variáveis
7 print("Nome:", nome)
8 print("Idade:", idade)
9 print("Altura:", altura)
10
```

# Variáveis e Tipos Embutidos

## 3 Para saber mais: Nomes de Variáveis

Os nomes de variáveis devem seguir certas regras:

1. Devem começar com uma letra ou um sublinhado (\_).
2. Podem conter letras, números e sublinhados.
3. Não podem ser palavras reservadas (keywords) de Python.

Exemplos de Nomes Válidos e Inválidos:

```
1 # Nomes válidos
2 _nome = "Bob"
3 idade2 = 25
4
5 # Nomes inválidos
6 2idade = 30 # Começa com um número
7 nome-id = "John" # Contém um caractere inválido '-'
```

# Variáveis e Tipos Embutidos

## 4 Instruções

Instruções são comandos que o Python executa. A maioria das instruções que você verá serão expressões, mas também existem instruções de controle de fluxo, como “if”, “for”, e “while”.

*Exemplo de Instruções Simples:*

```
1 # Instrução de atribuição
2 x = 10
3
4 # Instrução de impressão
5 print(x)
6
```

# Variáveis e Tipos Embutidos

## 5 Operadores Aritméticos

Python suporta operadores aritméticos para realizar cálculos matemáticos.

*Exemplos de Operadores Aritméticos:*

```
1 a = 10
2 b = 5
3
4 # Adição
5 soma = a + b
6 print("Soma:", soma) # 15
7
8 # Subtração
9 subtracao = a - b
10 print("Subtração:", subtracao) # 5
11
12 # Multiplicação
13 multiplicacao = a * b
14 print("Multiplicação:", multiplicacao) # 50
15
16 # Divisão
17 divisao = a / b
18 print("Divisão:", divisao) # 2.0
19
20 # Divisão inteira
21 divisao_inteira = a // b
22 print("Divisão Inteira:", divisao_inteira) # 2
23
24 # Módulo
25 modulo = a % b
26 print("Módulo:", modulo) # 0
27
28 # Exponenciação
29 exponenciacao = a ** b
30 print("Exponenciação:", exponenciacao) # 100000
31
```

# Variáveis e Tipos Embutidos

## 6 Strings

Strings são sequências de caracteres usadas para armazenar texto. Em Python, strings podem ser delimitadas por aspas simples ("") ou duplas ('').

*Exemplos de Operações com Strings:*

```
1 # Concatenar strings
2 saudacao = "Olá, " + "Mundo!"
3 print(saudacao) # Olá, Mundo!
4
5 # Repetir strings
6 eco = "Eco" * 3
7 print(eco) # EcoEcoEco
8
9 # Acessar caracteres individuais
10 primeira_letra = saudacao[0]
11 print(primeira_letra) # O
12
13 # Fatiar strings
14 parte = saudacao[4:9]
15 print(parte) # Mundo
16
```

# Variáveis e Tipos Embutidos

## 7 Entrada do Usuário

Para obter dados do usuário, utilizamos a função “`input()`”. Os dados retornados sempre serão do tipo `string`.

*Exemplo de Entrada de Usuário:*

```
1 nome = input("Qual é o seu nome? ")
2 print("Olá,", nome)
3
4 # Convertendo a entrada para outro tipo
5 idade = int(input("Quantos anos você tem? "))
6 print("Você tem", idade, "anos.")
7
```

# Variáveis e Tipos Embutidos

## 8 Constantes

Constantes são variáveis cujo valor não deve mudar ao longo do programa. Em Python, não há uma palavra-chave específica para declarar constantes, mas por convenção, usamos nomes em maiúsculas.

*Exemplo de Constantes:*

```
1 PI = 3.14159
2 GRAVIDADE = 9.8
3
4 print("PI:", PI)
5 print("Gravidade:", GRAVIDADE)
6
```

# Variáveis e Tipos Embutidos

## 9 Comando if

O comando “if” é usado para executar blocos de código com base em uma condição.

*Exemplo de Uso do Comando if:*

```
1 idade = 18
2
3 if idade >= 18:
4     print("Você é maior de idade.")
5 else:
6     print("Você é menor de idade.")
7
```

# Variáveis e Tipos Embutidos

## 10 Convertendo uma String para Inteiro

Muitas vezes, precisamos converter strings para outros tipos, como inteiros, para realizar operações matemáticas.

*Exemplo de Conversão de String para Inteiro:*

```
1 numero_str = "123"
2 numero_int = int(numero_str)
3 print(type(numero_int)) # <class 'int'>
4
```

# Variáveis e Tipos Embutidos

## 11 O comando elif

O comando “**elif**” (abreviação de “else if”) é usado para verificar múltiplas condições.

*Exemplo de Uso do Comando elif*

```
1 nota = 85
2
3 if nota >= 90:
4     print("A")
5 elif nota >= 80:
6     print("B")
7 elif nota >= 70:
8     print("C")
9 elif nota >= 60:
10    print("D")
11 else:
12     print("F")
13
```

# Variáveis e Tipos Embutidos

## 12 Exercícios - Jogo da Adivinhação

Vamos criar um simples jogo de adivinhação onde o jogador tenta adivinhar um número entre 1 e 10.

Código do Jogo da Adivinhação:

```
1 import random
2
3 numero_secreto = random.randint(1, 10)
4 tentativas = 0
5 acertou = False
6
7 while not acertou:
8     palpite = int(input("Adivinhe o número (entre 1 e 10): "))
9     tentativas += 1
10
11    if palpate == numero_secreto:
12        print("Parabéns! Você acertou em", tentativas, "tentativas.")
13        acertou = True
14    elif palpate < numero_secreto:
15        print("Tente um número maior.")
16    else:
17        print("Tente um número menor.")
```

# Variáveis e Tipos Embutidos

## 13 Comando while

O comando “while” é usado para repetir um bloco de código enquanto uma condição é verdadeira.

*Exemplo de Uso do Comando while:*

```
1 contador = 0
2
3 while contador < 5:
4     print("Contador:", contador)
5     contador += 1
6
```

# Variáveis e Tipos Embutidos

## 14 Exercícios - Jogo com while

Vamos modificar o jogo da adivinhação para limitar o número de tentativas a 3.

Código do Jogo com Limite de Tentativas:

```
1 import random
2
3 numero_secreto = random.randint(1, 10)
4 tentativas = 0
5 max_tentativas = 3
6 acertou = False
7
8 while tentativas < max_tentativas and not acertou:
9     palpito = int(input("Adivinhe o número (entre 1 e 10): "))
10    tentativas += 1
11
12    if palpito == numero_secreto:
13        print("Parabéns! Você acertou em", tentativas, "tentativas.")
14        acertou = True
15    elif palpito < numero_secreto:
16        print("Tente um número maior.")
17    else:
18        print("Tente um número menor.")
19
20 if not acertou:
21     print("Suas tentativas acabaram. O número era", numero_secreto)
22
```

# Variáveis e Tipos Embutidos

## 15 Comando for

O comando “**for**” é usado para iterar sobre uma sequência (como uma lista, tupla ou string).

*Exemplo de Uso do Comando for:*

```
1 # Iterando sobre uma lista
2 frutas = ["maçã", "banana", "cereja"]
3
4 for fruta in frutas:
5     print("Eu gosto de", fruta)
6
```

# Variáveis e Tipos Embutidos

## 16 Exercícios - Utilizando o *for* no jogo

Vamos criar um jogo de adivinhação que dá três chances ao usuário para adivinhar o número.

*Código do Jogo usando for:*

```
1 import random
2
3 numero_secreto = random.randint(1, 10)
4 max_tentativas = 3
5
6 for tentativas in range(max_tentativas):
7     palpite = int(input("Adivinhe o número (entre 1 e 10): "))
8
9     if palpate == numero_secreto:
10         print("Parabéns! Você acertou em", tentativas + 1, "tentativas.")
11         break
12     elif palpate < numero_secreto:
13         print("Tente um número maior.")
14     else:
15         print("Tente um número menor.")
16 else:
17     print("Suas tentativas acabaram. O número era", numero_secreto)
18
```

# Conclusão

Este guia abrangeu os fundamentos básicos de Python, incluindo variáveis, tipos de dados, operadores, estruturas de controle e entrada de usuário. Cada um desses tópicos foi apresentado com exemplos práticos para facilitar seu entendimento e aplicação.

Ao longo deste eBook, você aprendeu como declarar e usar variáveis, realizar operações matemáticas, manipular strings, interagir com o usuário e controlar o fluxo de seu programa com condicionais e loops. Além disso, exploramos como criar um simples jogo de adivinhação, aplicando os conceitos discutidos.

Agora que você possui uma base sólida em Python, o próximo passo é praticar. Experimente os exemplos, modifique-os e tente criar seus próprios programas. A prática constante é essencial para consolidar seu aprendizado e ganhar confiança na programação.

Lembre-se de que a programação é uma habilidade prática que se aprimora com o tempo e a experiência. Não tenha medo de cometer erros e sempre busque entender o que deu errado para aprender com essas experiências. Existem muitos recursos online, como tutoriais, fóruns e comunidades, que podem oferecer suporte e inspiração.

Boa sorte em sua jornada de programação! Divirta-se explorando o mundo da programação com Python, e continue aprendendo e desafiando a si mesmo. O céu é o limite!

# Contato com o Autor

Obrigado por ler o "Guia Completo de Python para Iniciantes"! Este eBook foi criado com o objetivo de fornecer uma introdução sólida e prática ao mundo da programação em Python. Se você tiver dúvidas, sugestões ou apenas quiser compartilhar suas experiências, sinta-se à vontade para entrar em contato.

## *Informações de Contato:*

*GitHub : <https://github.com/eduardo-ferreira-f0>*

## *Agradecimentos*

Este eBook foi criado como parte de um projeto do bootcamp da DIO (Digital Innovation One). A organização e criação do conteúdo foram feitas utilizando o Canvas, uma ferramenta excelente para design e apresentação.

- *ChatGPT: Utilizado para auxiliar na criação do roteiro e conteúdo do eBook.*
- *Leonardo AI: Responsável pela arte da capa do eBook.*

Agradecemos a todas as ferramentas e plataformas que contribuíram para tornar este projeto possível.

Desejo a você uma jornada de aprendizado produtiva e divertida com Python. Boa sorte e continue explorando o maravilhoso mundo da programação!

Espero que este eBook tenha sido útil e inspirador para você. Vamos continuar aprendendo e crescendo juntos!

Atenciosamente,  
Eduardo Ferreira.

