

# Glossário Completo de Python para Iniciantes

Este glossário foi desenvolvido para quem deseja **começar a programar em Python**, mesmo que ainda **não saiba nada**. Você vai aprender os principais conceitos, estruturas e exemplos práticos de código.

---

## ★ O que é Python?

Python é uma linguagem de programação **fácil de aprender, versátil** e muito utilizada em áreas como **automação, ciência de dados, desenvolvimento web, inteligência artificial** e muito mais.

Exemplo básico:

```
print("Olá, mundo!")
```

---

## ★ Variáveis

Declaração de variáveis **não precisa definir o tipo**:

```
nome = "Ana"  
idade = 25  
altura = 1.70
```

---

## ★ Tipos de Dados

Tipo	Exemplo
int	10, -5, 1000
float	3.14, -0.01
str	"Texto"
bool	True, False
list	[1, 2, 3]
tuple	(1, 2, 3)
dict	{"nome": "Ana"}
set	{1, 2, 3}

---

## ★ Operadores

Tipo	Exemplo
Aritméticos	+, -, *, /, %, **, //
Comparação	==, !=, >, <, >=, <=
Lógicos	and, or, not

---

## ✦ Entrada de Dados

```
nome = input("Digite seu nome: ")  
print("Olá,", nome)
```

---

## ✦ Condicionais

```
idade = 18  
  
if idade >= 18:  
    print("Maior de idade")  
else:  
    print("Menor de idade")
```

---

## ✦ Laços de Repetição

**For:**

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

**While:**

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

---

## ✦ Listas

```
frutas = ["Maçã", "Banana", "Uva"]  
print(frutas[0])
```

Adicionar elemento:

```
frutas.append("Pera")
```

---

## ✦ Tuplas

```
cores = ("Vermelho", "Verde", "Azul")
```

---

## ✦ Dicionários

```
pessoa = {"nome": "Ana", "idade": 30}
print(pessoa["nome"])
```

---

## ✦ Funções

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

print(saudacao("Carlos"))
```

---

## ✦ Funções Lambda (Anônimas)

```
soma = lambda x, y: x + y
print(soma(2, 3))
```

---

## ✦ Manipulação de Strings

```
texto = "Python é incrível"
print(texto.upper())
print(texto.lower())
print(texto.replace("incrível", "fácil"))
```

---

## ✦ Manipulação de Arquivos

```
with open("arquivo.txt", "w") as arquivo:
    arquivo.write("Olá, arquivo!")
```

---

## ✦ Tratamento de Erros

```
try:
    numero = int(input("Digite um número: "))
except ValueError:
    print("Valor inválido!")
```

---

## ✦ Classes e Objetos (Programação Orientada a Objetos)

```
class Pessoa:
    def __init__(self, nome):
        self.nome = nome

    def falar(self):
        print(f"Olá, meu nome é {self.nome}")

p1 = Pessoa("Carlos")
p1.falar()
```

---

## ✦ List Comprehension

```
quadrados = [x**2 for x in range(5)]
```

---

## ✦ Módulos e Bibliotecas

Importando bibliotecas externas:

```
import math
print(math.sqrt(16))
```

Instalando bibliotecas:

```
bash
pip install requests
```

---

## ✦ Funções Assíncronas (async/await)

```
import asyncio

async def main():
    print("Olá")
    await asyncio.sleep(1)
    print("Depois de 1 segundo")

asyncio.run(main())
```

---

## ✦ Boas Práticas

- ☐ Usar nomes claros para variáveis e funções
  - ☐ Seguir o estilo **PEP 8** (indentação, nomes, etc.)
  - ☐ Documentar o código
  - ☐ Dividir código em funções e módulos
-

## ✦ Aplicações Comuns em Python

- Automação
- Web Scraping
- Desenvolvimento Web (Django, Flask)
- Análise de Dados
- Inteligência Artificial