Python

Considerada linguagem de alto nível

Pacote: “C” 🡪 “Assemble”

DOC > PY = 1000100 > C > 00000001 > ASSEMBLE 🡪 Máquina

Print(“Hello Word!”)

Interpretador

<https://wiki.python.org/moin/BeginnersGuide>

<https://devguide.python.org/>

<https://pypy.org/>

<https://micropython.org/>

<https://thonny.org/>

<https://discuss.python.org/t/improving-python-language-reference-plr-documentation/36709>

<https://python.org.br/>

<https://pypi.org/>

Alguns especialistas no assunto de IA:  
>Augusto Galego; Lucas Montano; Arthur Igreja; Mano Deivyn; Jovem Tranquilão; Atekita; Fabio Akita; Felipe Deschamp; Micael Mota; Codigo Fonte TV; Guanabara; LinuxTips; Jefferson Fernando(BAIDUX)

google colob 🡪 IDE online com python nativo

replit 🡪 <https://replit.com/languages/python3> --> IDE nuvem python nativo

gitpod 🡪 <https://www.gitpod.io/> 🡪 IDE nuvem

vscode 🡪 IDE editor de código

pycharm 🡪 IDE editor python para dados

Universo Python

py 🡪 fastapi 🡪 flask 🡪 django 🡪 streamlite

22/04/25

Python: é uma linguagem de programação de alto nível, interpretada, de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte. Foi lançada por Guido van Rossum em 1991 e desenvolvida por uma comunidade que determina seu direcionamento. A filosofia de Python enfatiza a legibilidade do código com sua notável sintaxe clara. É uma linguagem que permite expressar conceitos em menos linhas de código do que seria possível em linguagens como C++ ou Java. Esta simplicidade torna Python uma excelente escolha tanto para iniciantes quanto para desenvolvedores experientes. Python é versátil e encontra aplicações em diversos campos:

• Desenvolvimento web (Django, Flask)

• Ciência de dados e análise estatística

• Inteligência artificial e machine learning

• Automação e scripting

• Desenvolvimento de jogos

• Aplicações desktop

Uma das maiores vantagens do Python é seu ecossistema rico em bibliotecas e frameworks que facilitam o desenvolvimento em praticamente qualquer área.

🡪Repositorios(Tech); 🡪IDE’s(Tech); 🡪Versionamentos(Tech);

\*GitHub \* \*Git

\*GitLab \* \*

\*BitBucket \* \*

GitHub Desktop

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_23/04/25

print("Hello, World!")  # Exibe a mensagem "Hello, World!" na tela

print("Olá, Mundo!")  # Exibe a mensagem "Olá, Mundo!" na tela

# Criando variáveis

nome = "Gilmar"

idade = 46

altura = 1.61

is\_estudante = True

# Exibindo os valores

print(nome)  # Saída: Gilmar

print(idade)  # Saída: 46

print(altura)  # Saída: 1.61

print(is\_estudante)  # Saída: True

print(nome, idade, altura, is\_estudante)  # Saída: Gilmar 46 1.61 True

# 'str' string: textos, caracteres, letras, etc. "Gilmar"

# 'int' inteiro: números inteiros, positivos ou negativos "46"

# 'float' ponto flutuante: números decimais, positivos ou negativos "1.61"

# 'bool' booleano: verdadeiro ou falso "True" ou "False"

idade = 25

temperatura\_negativa = -10

numero\_grande = 1000000

# Em Python 3, não há limite prático para o tamanho dos inteiros

numero\_muito\_grande = 10\*\*100  # 10 elevado a 100

print(idade)  # Saída: 25

print(temperatura\_negativa)  # Saída: -10

print(numero\_grande)  # Saída: 1000000

print(numero\_muito\_grande)  # Saída: 100

# Entrada de dados

# input() - lê uma linha de entrada do usuário e retorna como string

nome = input("Digite seu nome: ")  # Lê o nome do usuário

idade = input("Digite sua idade: ")  # Lê a idade do usuário

print("Seu nome é:", nome)  # Exibe o nome do usuário

print("Sua idade é:", idade)  # Exibe a idade do usuário

idade = int(input("Digite sua idade: "))  # Lê a idade do usuário e converte para inteiro

print("Sua idade é:", idade)  # Exibe a idade do usuário

quilo = float(input("Digite seu peso em quilos: "))  # Lê o peso do usuário e converte para float

print("Seu peso é:", quilo)  # Exibe o peso do usuário

# Operadores Matemáticos

# +, -, \*, /, //, %, \*\*

# + - soma

# - - subtração

# \* - multiplicação

# / - divisão

# // - divisão inteira

# % - resto da divisão

# \*\* - potência

# Exemplo de uso dos operadores matemáticos

# a = 10

# b = 3

# soma = a + b  # Soma

# subtracao = a - b  # Subtração

# multiplicacao = a \* b  # Multiplicação

# divisao = a / b  # Divisão

# divisao\_inteira = a // b  # Divisão inteira

# resto\_divisao = a % b  # Resto da divisão

# potencia = a \*\* b  # Potência

# print("Soma:", soma)  # Exibe o resultado da soma

# print("Subtração:", subtracao)  # Exibe o resultado da subtração

# print("Multiplicação:", multiplicacao)  # Exibe o resultado da multiplicação

# print("Divisão:", divisao)  # Exibe o resultado da divisão

# print("Divisão inteira:", divisao\_inteira)  # Exibe o resultado da divisão inteira

# print("Resto da divisão:", resto\_divisao)  # Exibe o resultado do resto da divisão

# print("Potência:", potencia)  # Exibe o resultado da potência

a=10

b=3

print(a+b)  # Soma

print(a-b)  # Subtração

print(a\*b)  # Multiplicação

print(a/b)  # Divisão

print(a//b)  # Divisão inteira

print(a%b)  # Resto da divisão

print(a\*\*b)  # Potência

# Condicionais (if, elif, else)

# if - se

# elif - se não, se

# else - se não

idade = int(input("Digite sua idade: ?"))  # Lê a idade do usuário e converte para inteiro

if idade >= 18:  # Se a idade for maior ou igual a 18

    print("Você é maior de idade.")  # Exibe mensagem se a condição for verdadeira

elif idade > 12:  # Se a idade for maior a 12 e menor que 18

    print("Você é adolescente.")  # Exibe mensagem se a condição for verdadeira

else:  # Se a idade for menor que 12

    print("Você é criança.")

# Exibe mensagem se a condição for verdadeira

# Laços de Repetição (for, while)

# for - para

# while - enquanto

# Exemplo de uso do laço for

# for i in range(5):  # Laço que itera de 0 a 4

#     print(i)  # Exibe o valor de i a cada iteração

# Exemplo de uso do laço while

# i = 0  # Inicializa a variável i

# while i < 5:  # Enquanto i for menor que 5

#     print(i)  # Exibe o valor de i

#     i += 1  # Incrementa i em 1 a cada iteração

# Exibe o valor de i a cada iteração

contador = 0  # Inicializa o contador

while contador < 5:  # Enquanto o contador for menor que 5

    print("Contador:", contador)  # Exibe o valor do contador

    contador += 1  # Incrementa o contador em 1 a cada iteração

for i in range(5):  # Laço que itera de 0 a 4

    print("Valor de i:", i)  # Exibe o valor de i a cada iteração

    # Exibe o valor de i a cada iteração

echo "# python\_aulas\_john" >> README.md

git init

git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/GilmarCz/python\_aulas\_john.git

git push -u origin main

git push origin main