

Modulo 1 - Introducao

Preparacao

Porque aprender Python?

1. Facilidade de uso: Python é conhecido por sua sintaxe simples e fácil de ler, o que o torna acessível para iniciantes e experientes.
2. Ampla adoção na indústria: Python é amplamente utilizado em diferentes setores da indústria devido à sua versatilidade e eficácia em lidar com uma variedade de tarefas relacionadas a dados.
3. Ecossistema de bibliotecas: Python possui um rico ecossistema de bibliotecas para análise de dados, como pandas, NumPy e matplotlib, que facilitam a manipulação e visualização de dados.
4. Ferramentas de visualização de dados: Com bibliotecas como matplotlib, seaborn e plotly, é possível criar visualizações de dados interativas e informativas para comunicar resultados de análises.
5. fácil integração: Python pode ser facilmente integrado com outras tecnologias e sistemas, tornando-o uma escolha ideal para administradores de dados que precisam trabalhar com diferentes ferramentas e plataformas.
6. Aprendizado de máquina e análise de dados: Python é a linguagem de escolha para muitos profissionais que trabalham com aprendizado de máquina e análise de dados, devido às suas bibliotecas poderosas, como scikit-learn, TensorFlow e PyTorch.
7. Automação de tarefas: Python pode ser usado para automatizar tarefas repetitivas, como limpeza de dados, geração de relatórios e processamento de arquivos, economizando tempo e aumentando a eficiência.

Baixando Itens Necessarios e Boas Referencias

1. [Vs code](#)
2. [Git Hub Desktop](#)
3. [Criar uma conta no GitHub](#)

Conceitos Iniciais

Nessa parte iremos fazer um breve resumo dos conceitos mais importantes para se estabelecer uma comunicacao eficiente entre o programador e a maquina. Dito isso, cabe a nos entender como um processo de comunicao como qualquer outra lingua falada.

1. Qual é o codigo que estamos usando?(tipos de dados)
2. Como se comportar como emissor(input de dados)
3. Como ser o receptor(output de dados)
4. Como podemos manipular esses dados no inicio(Operadores Aritimeticos)

Tipos de dados

Definicao Informal mas eficiente!

- String(str): Conjunto de caracteres - Criadas com "" ou "
 - ex: "Motim" ou "Educacao"
- Integer(int): Numeros inteiros
 - ex: 2, 44
- Float(float): Numeros de **ponto** flutuante
 - ex: 2.0, 44.343
- Boolean(bool): Logica binaria de True or False / 1 ou 0

Curiosidade: [George Boole](#) (Lincoln, 2 de novembro de 1815 — Ballintemple, 8 de dezembro de 1864) foi um matemático, filósofo britânico, criador da [álgebra booleana](#), fundamental para o desenvolvimento da computação moderna.

Operadores Aritmeticos

Operador	Simbolo
Soma	+
Subtracao	-
Multiplicacao	*
Exponenciacao	**
Divisao	/
Divisao Inteira	//
Resto	%

Output Terminal - print

Em Python, a função print() é usada para exibir saídas no terminal. Você pode passar uma string como argumento para a função print() para exibi-la no terminal. Por exemplo:

```
In [ ]: print("Olá, mundo!")
```

Isso exibirá "Olá, mundo!" no terminal quando o código for executado. Você também pode passar várias strings separadas por vírgulas para a função `print()` para exibi-las em sequência:

```
In [ ]: nome = "Alice"
idade = 30
print("Olá,", nome, "você tem", idade, "anos.")
```

Isso exibirá "Olá, Alice você tem 30 anos." no terminal. Além disso, você pode formatar a saída usando f-strings (a partir do Python 3.6):

```
In [ ]: nome = "Bob"
idade = 25
print(f"Olá, {nome}, você tem {idade} anos.")
```

Input - Terminal

Em Python, você pode receber entrada do usuário no terminal usando a função `input()`. Esta função solicita ao usuário que insira um valor e retorna o que foi digitado como uma string. Aqui está um exemplo simples:

```
In [ ]: nome = input("Qual é o seu nome? ")
print("Olá,", nome)
```

Neste exemplo, o programa solicitará ao usuário que insira seu nome. O valor digitado pelo usuário será armazenado na variável `nome` e, em seguida, o programa imprimirá "Olá, " seguido do nome digitado pelo usuário.

É importante notar que o **valor retornado por `input()` é sempre uma string**. Se você precisar de um tipo diferente (como um número inteiro), será necessário converter o valor usando as funções apropriadas (por exemplo, `int()` para inteiros).

Pratique

- Criar conta e finalizar login: <https://judge.beecrowd.com/pt/problems/index/1>
- Problemas recomendados: 1000 ate 1008
- **1009/1012**

Condicionais

Sao estruturas de controle de fluxo condicionais `if`, `elif` (abreviação de "else if") e `else` para executar diferentes blocos de código com base em condições específicas.

Quando a condicao for verdadeira o codigo seguira o escopo desse if. Aqui está um exemplo simples:

```
In [ ]: idade = int(input("Quantos anos você tem? "))

if idade < 18:
    print("Você é menor de idade.")
elif idade >= 18 and idade < 60:
    print("Você é adulto.")
else:
    print("Você é idoso.")
```

Neste exemplo, o programa solicita ao usuário que insira sua idade e armazena o valor como um número inteiro na variável idade. Em seguida, o programa verifica a idade e imprime uma mensagem com base na faixa etária.

- A estrutura if verifica se a idade é menor que 18 e imprime "Você é menor de idade." se for verdadeiro.
- A estrutura elif verifica se a idade é maior ou igual a 18 e menor que 60 e imprime "Você é adulto." se for verdadeiro.
- A estrutura else imprime "Você é idoso." se nenhuma das condições anteriores for verdadeira.

Você pode usar quantas cláusulas elif forem necessárias para abranger todas as condições desejadas. A cláusula else é opcional e pode ser omitida se não houver necessidade de um bloco de código para todas as outras condições.

Outro exemplo essencial:

```
In [ ]: frase = 'motim educacao - python'

if 'python' in frase:
    print("Python está na frase.")
if 'motim' in frase:
    print("Motim está na frase.")
if 'educacao' in frase:
    print("Educação está na frase.")
```

segunda versao:

```
In [ ]: frase = 'motim educacao - python'

if 'python' in frase:
    print("Python está na frase.")
elif 'motim' in frase:
    print("Motim está na frase.")
elif 'educacao' in frase:
    print("Educação está na frase.")
```

Pratique

- 1035,1038,1047

- 1049,1040,1046