|  |  |  |
| --- | --- | --- |
|  | **PYTHON PARA CIÊNCIA DE DADOS**  **STRINGS** | Imagem relacionada |

**Definição**: variáveis do tipo *string* armazenam sequências de caracteres, tais como nomes e textos em geral. Trata-se de uma sequência de símbolos, como letras, números, sinais de pontuação, etc. Uma *string* também é uma lista, ou seja, um conjunto ordenado de elementos que podem ser acessados por índices. Só que, diferentemente de listas em geral, *strings* contêm sempre elementos do mesmo tipo: o tipo **caractere**. A manipulação de listas e strings é feita da mesma maneira.

**Exemplo**: uma *string* chamada ‘teste’ que recebe o texto ‘bom dia’.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| teste = | ‘b’ | ‘o’ | ‘m’ |  | ‘d’ | ‘i’ | ‘a’ |
|  | [0] | [1] | [2] | [3] | [4] | [5] | [6] |

|  |  |  |
| --- | --- | --- |
| *# criando a string*    *# obtendo o tamanho da string*    *# acessando a string inteira*    *# concatenando a string com outro texto* | *# acessando o primeiro caractere da string*    *# acessando o último caractere da string*    *# acessando partes da string* | *# percorrendo a string inteira* |

**EXERCÍCIOS**

1. Criar um algoritmo que lê do usuário uma expressão matemática qualquer que faz uso de parênteses. O programa deve verificar e imprimir na tela se a expressão está correta com relação ao uso dos parênteses, ou seja, verificar se não há parênteses sobrando ou faltando. Veja abaixo exemplos de execução:

|  |  |  |
| --- | --- | --- |
|  |  |  |

1. O programa em Python abaixo lê uma string do usuário, apresentando na tela algumas informações sobre a mesma.

|  |  |
| --- | --- |
| **Programa em Python** | **Exemplos de execução desejada** |
| **# -- definição das funções –**  # os estudantes devem completar essa parte  **# -- programa principal --**  # entrada de dados  texto = input('Digite uma palavra ou frase:')  # obtenção dos resultados  tce = ContCaracTotal(texto)  tse = ContCaracSemEsp(texto)  vog = ContVogais(texto)  con = ContConsoantes(texto)  # impressão dos resultados  print('Total de caracteres.................', tce)  print('Total de caracteres (sem espaços)...', tse)  print('Quantidade de vogais................', vog)  print('Quantidade de consoantes............', con) |  |

Faça o que se pede:

1. Copie o código fornecido e cole em algum editor de sua preferência (Spyder, Pycharm, Idle, etc.).
2. Defina a função **ContCaracTotal(texto)**, que retorna a quantidade total de caracteres da string passada como parâmetro.
3. Defina a função **ContCaracSemEsp(texto)**, que retorna a quantidade de caracteres da string passada como parâmetro desconsiderado os espaços.
4. Defina a função **ContVogais(texto)**, que retorna a quantidade de vogais existentes na string passada como parâmetro.
5. Defina a função **ContConsoantes(texto)**, que retorna a quantidade de consoantes existentes na string passada como parâmetro.
6. O programa em Python abaixo lê uma data de nascimento do usuário no formato dd/mm/aaaa, imprimindo na tela o mês (por extenso) do aniversário.

|  |  |
| --- | --- |
| **Programa em Python** | **Exemplos de execução desejada** |
| **# -- definição da função --**  # os estudantes devem completar essa parte  **# -- programa principal --**  data = input('Digite sua data de nasc. (dd/mm/aaaa):')  mes\_nasc = ObtemMesNascimento(data)  print('Você nasceu em:',mes\_nasc) |  |

Faça o que se pede:

1. Copie o código fornecido e cole em algum editor de sua preferência (Spyder, Pycharm, Idle, etc.).
2. Defina a função **ObtemMesNascimento(data)**, que retorna o mês de nascimento escrito por extenso.