

Lista de exercícios -LP-17

Exercícios: tipo de dados STRING (parte 1)

Para cada um dos exercícios, crie um arquivo fonte Python com o respectivo nome de acordo com as regras já determinadas nas listas de exercícios anteriores.

DICA: muitas vezes um mesmo exercício com strings pode ser resolvido de diferentes maneiras, todas elas corretas. Você pode usar laços com **for** e/ou **while**, **indexação** e **fatiamento** de strings etc. Use sua criatividade e seu raciocínio lógico para encontrar soluções que façam mais sentido para você. Bom trabalho!

Questões:

1. Escreva um programa Python que leia uma string e um caractere e diga (mostre) se o caractere aparece pelo menos uma vez na string.
2. Escreva um programa Python que leia uma string e um caractere e diga (mostre) quantas vezes o caractere aparece na string.
3. Faça um programa que solicite o nome do usuário e imprima-o na vertical. Exemplo: FULANO.

```
F
U
L
A
N
O
```

4. Modifique o programa anterior de forma a mostrar o nome em formato de escada. Exemplo: FULANO.

```
F
FU
FUL
FULA
FULAN
FULANO
```

5. Altere o programa anterior de modo que a escada seja invertida. Exemplo: FULANO

```
FULANO
FULAN
FULA
FUL
FU
F
```

6. Escreva um programa Python que leia uma string e um caractere, e depois crie e imprima uma outra string sem o caractere lido.
7. Faça um programa Python que leia uma string e crie uma outra string repetindo apenas as vogais.
Ex: carro => caarroo
8. Faça um programa Python que leia 2 strings e informe o conteúdo delas seguido do seu comprimento. Informe também se as duas strings possuem o mesmo comprimento e são iguais ou diferentes no conteúdo.

Exemplo:

```
Compara duas strings
String 1: Brasil Hexa 2006
String 2: Brasil! Hexa 2006!
Tamanho de "Brasil Hexa 2006": 16 caracteres
Tamanho de "Brasil! Hexa 2006!": 18 caracteres
As duas strings são de tamanhos diferentes.
As duas strings possuem conteúdo diferente.
```

9. Dado uma string com uma frase informada pelo usuário (incluindo espaços em branco), conte e imprima: quantos espaços em branco existem na frase; e quantas vezes aparecem as vogais a, e, i, o, u.
10. Faça um programa Python que solicite a data de nascimento (dd/mm/aaaa) do usuário e imprima a data com o nome do mês por extenso. Exemplo:

```
Data de Nascimento: 29/10/1973
Você nasceu em 29 de Outubro de 1973.
```

11. Faça um programa Python que solicite a digitação de um número de CPF **no formato xxx.xxx-xx** e indique se é um número válido ou inválido através da validação dos dígitos verificadores e dos caracteres de formatação.

Exemplos:

```
O CPF 123.456.789-12 é inválido!
O CPF 529.982.247-25 é válido!
O CPF 777.777.777-77 é inválido!
```

Observação: a validade do CPF é feita com cálculos simples utilizando os 9 primeiros dígitos e comparando o resultado com os 2 últimos dígitos. **Pesquise na internet as regras para validação de CPF.**