



**Faculdade Senac**

**Curso: Análise e Desenvolvimento de Sistemas | UC: Coding e Algoritmos**

**Docente: Tereza Oliveira - terezacoliveira@senac.pe.br**

## Desafios Python 2.0

Olá! Nesta tarefa, você completará uma série de desafios que se concentram na manipulação de strings e caracteres em Python. Cada desafio tem um objetivo específico e dicas sobre métodos que você pode usar para alcançar o resultado desejado. Siga as instruções detalhadas abaixo para cada desafio, para garantir que você entenda como completá-los com sucesso.

Clique aqui para aprender sobre: [Função Input e Print](#), [Métodos de String em Python](#), [Variáveis em Python](#).

### ♦ Desafio 1: Contando Caracteres

#### Objetivo

Sua tarefa é pedir ao usuário para inserir uma frase e, em seguida, calcular quantos caracteres existem nessa frase, excluindo quaisquer espaços.

#### Etapas para concluir

1. **Avisar o usuário:** Use a função 'input()' para pedir ao usuário que digite uma frase.
2. **Remover espaços:** Utilize o método '.replace()' para eliminar espaços da frase. Por exemplo, você pode substituir espaços por uma cadeia de caracteres vazia.
3. **Contar caracteres:** Use a função 'len()' para contar o número de caracteres na string modificada (aquela sem espaços).
4. **Exibir o resultado:** Imprima o número total de caracteres para o usuário.

#### Exemplo

Se o usuário inserir "Hello World", seu programa deverá gerar 10 caracteres.



## ♦ Desafio 2: Todas as letras maiúsculas

### Finalidade

Neste desafio, você converterá uma cadeia de caracteres fornecida pelo usuário em letras maiúsculas.

### Etapas para concluir

1. **Obter entrada do usuário:** Peça ao usuário para inserir uma string usando a função `'input()'`.
2. **Converter para maiúsculas:** Aplique o método `'upper()'` à string para alterar todas as letras para maiúsculas.
3. **Saída do resultado:** Imprima a versão em maiúsculas da string para o usuário ver.

### Exemplo

Se o usuário digitar "hello", a saída deverá ser "HELLO".

## ♦ Desafio 3: Verificando Palíndromos

### Objetivo

Seu objetivo é pedir ao usuário uma palavra e determinar se é um palíndromo (uma palavra que lê o mesmo para trás e para frente).

### Etapas para concluir

1. **Request Input:** Use `'input()'` para obter uma palavra do usuário.
2. **Normalizar a palavra:** Converta a palavra em minúsculas usando `'lower()'` para garantir que a verificação não diferencie maiúsculas de minúsculas.
3. **Verifique se há palíndromo:** Use o fatiamento `('[::-1]')` para inverter a string e compará-la com a original.
4. **Exibir o resultado:** Informe ao usuário se a palavra é um palíndromo ou não.

### Exemplo

Para a entrada "Radar", a saída deve indicar que é um palíndromo.



## ♦ Desafio 4: Contador de Vogais

### Objetivo

Conte o número de vogais em uma cadeia de caracteres fornecida pelo usuário.

### Etapas para concluir

1. **Obter entrada:** Solicita que o usuário insira uma cadeia de caracteres.
2. **Normalizar a string:** Converta a string em minúsculas usando `'.lower()'`.
3. **Inicialize um contador:** Crie uma variável para acompanhar o número de vogais.
4. **Loop Through the String:** Use um loop `'for'` para iterar sobre cada caractere na string.
5. **Verifique se há vogais:** Dentro do loop, verifique se cada caractere é uma vogal (a, e, i, o, u). Se estiver, incremente seu contador.
6. **Output the Count:** Após o loop, imprima o número total de vogais encontradas.

### Exemplo

Se o usuário inserir "Hello World", a saída deverá indicar que há 3 vogais.

## ♦ Desafio 5: Substituindo palavras

### Objetivo

Você pedirá ao usuário uma frase e substituirá uma palavra específica por outra palavra de sua escolha.

### Etapas para concluir

1. **Prompt for Input:** Use `'input()'` para obter uma frase do usuário.
2. **Peça palavras:** Solicite que a palavra seja substituída e a nova palavra pela qual substituí-la.
3. **Executar substituição:** Use o método `'.replace()'` para substituir a palavra especificada na frase.
4. **Exibir a nova frase:** Imprima a frase modificada para o usuário.

### Exemplo

Se o usuário inserir "Eu amo maçãs" e quiser substituir "maçãs" por "laranjas", a saída deverá ser "Eu amo laranjas".



## ♦ Desafio 6: Nome formatado

### Finalidade

Neste desafio, você pedirá o nome completo do usuário e o exibirá em vários formatos.

### Etapas para concluir

1. **Obter nome completo:** Use `'input()'` para solicitar ao usuário seu nome completo.
2. **Converter para maiúsculas:** Use `'.upper()'` para exibir o nome em maiúsculas.
3. **Converter para minúsculas:** Use `'.lower()'` para exibir o nome em letras minúsculas.
4. **Contar letras totais:** Use `'len()'` no nome depois de remover os espaços com `'replace()'` para encontrar o número total de letras.
5. **Extrair nome:** Use `'.split()'` para separar o primeiro nome do resto do nome e conte suas letras usando `'len()'`.
6. **Exibir todos os resultados:** Imprima cada um dos formatos e contagens para o usuário.

### Exemplo

Para a entrada "John Doe", a saída deve mostrar:

- Todas as letras maiúsculas: "JOHN DOE"
- Todas as minúsculas: "john doe"
- Total de letras: 8
- Nome: "John" (4 letras)

## ♦ Desafio 7: Invertendo a String

### Finalidade

Você pedirá ao usuário para digitar uma frase e, em seguida, exibi-la na ordem inversa.

### Etapas para concluir

1. **Request Input:** Use `'input()'` para obter uma frase do usuário.
2. **Inverter a frase:** Use fatiamento (`'[::-1]'`) para inverter a string.
3. **Saída da frase invertida:** Imprima a frase invertida para o usuário ver.

### Exemplo

Se o usuário digitar "Hello World", a saída deverá ser "dlroW olleH".



## ♦ Desafio 8: Verificando a substring

### Finalidade

Neste desafio, você verificará se uma palavra específica está contida em uma frase.

### Etapas para concluir

1. **Obter entrada de frase:** Peça ao usuário para inserir uma frase.
2. **Obter palavra para verificar:** Solicite ao usuário a palavra que deseja verificar.
3. **Verificar substring:** Use a palavra-chave 'in' para ver se a palavra existe na frase, convertendo ambas em minúsculas com '.lower()' para uma verificação que não diferencia maiúsculas de minúsculas.
4. **Exibir o resultado:** Informe ao usuário se a palavra é encontrada na frase.

### Exemplo

Se o usuário inserir "O gato está no telhado" e verificar se há "gato", a saída deverá confirmar que a palavra está presente.

## ♦ Desafio 9: CPF formatado

### Finalidade

Você pedirá um número de CPF (apenas os dígitos) e o formatará em uma estrutura específica.

### Etapas para concluir

1. **Solicitar entrada:** Use 'input()' para obter o número do CPF do usuário.
2. **Formate o CPF:** Use fatiamento e concatenação para reorganizar os dígitos no formato '000.000.000-00'.
3. **Exibir o CPF Formatado:** Imprima o número do CPF formatado para o usuário.

### Exemplo

Se o usuário inserir "12345678901", a saída deverá ser "123.456.789-01".

## ♦ Desafio 10: Soma dos Dígitos



## Objetivo

Sua tarefa é pedir um número inteiro e calcular a soma de todos os seus dígitos.

## Etapas para concluir

1. **Obter entrada numérica:** Use 'input()' para solicitar ao usuário um número inteiro.
2. **Converter em String:** Use 'str()' para converter o número em uma string para que você possa iterar sobre cada dígito.
3. **Inicialize uma variável de soma:** Crie uma variável para acompanhar a soma total dos dígitos.
4. **Loop Through Each Digit:** Use um loop 'for' para percorrer cada caractere na representação da string do número.
5. **Converter e adicionar:** Dentro do loop, converta cada caractere de volta em um inteiro usando 'int()' e adicione-o à sua variável de soma.
6. **Exibir o Total:** Após o loop, imprima a soma total dos dígitos para o usuário.

## Exemplo

Se o usuário inserir "123", a saída deve indicar que a soma dos dígitos é 6.