

### Exercício 1: Contando Palavras

Escreva um programa que leia uma frase do usuário e diga quantas vezes a palavra "Python" aparece na frase, independentemente de ser maiúscula ou minúscula.

Dica: Use os métodos `lower()` e `count()`.

### Exercício 2: Verificando Início e Fim

Peça ao usuário para digitar uma frase. Verifique se:

- A frase **começa com** "Bom dia"
- A frase **termina com** "obrigado"

Mostre mensagens diferentes dependendo do resultado.

### Exercício 3: Encontrando Palavras

Solicite uma frase ao usuário e pergunte por uma palavra a ser procurada.

Informe se a palavra **existe** na frase e, se existir, em **qual posição ela aparece pela primeira vez**.

### Exercício 4: Censurando Palavras

Escreva um programa que substitua todas as ocorrências da palavra "bobo" por "\*\*\*\*" em uma frase digitada pelo usuário.

### Exercício 5: Trocando letras

Peça ao usuário para digitar uma palavra.

Substitua todas as letras "a" por "@" e "e" por "3", e exiba o resultado.

### Exercício 6: Contando vogais

Peça uma frase ao usuário e mostre quantas vezes cada vogal (a, e, i, o, u) aparece.

**Dica:** Use `count()` para cada vogal após converter a frase para minúsculas com `lower()`.

### Exercício 7: Palavra invertida

Peça uma palavra e mostre-a **de trás para frente**.

Depois diga se ela é um **palíndromo** (lê igual de trás pra frente).

Dica: Use slicing `[::-1]`.

### Exercício 8: Validador de senha simples

Peça ao usuário que digite uma senha.

Verifique se a senha:

- Tem **pelo menos 8 caracteres**
- Contém **pelo menos um caractere especial** (como !@#\$%&\*)

Se a senha for válida, exiba "Senha aceita".

Caso contrário, informe o que está faltando.