

Recapitulação de Programação com Python II

As estruturas de repetição (ou loops) são fundamentais em Python para automatizar tarefas repetitivas. Em vez de escrever o mesmo código várias vezes, você instrui o computador a executar um bloco de código enquanto uma condição for verdadeira ou para cada item de uma sequência.

Em Python, temos dois tipos principais: `for` e `while`.

1. `for`

O `for` é usado quando você sabe antecipadamente quantas vezes deseja repetir algo ou quando quer percorrer os elementos de uma coleção (como uma **lista, string ou dicionário**).

Exemplo 01 - Vamos criar um programa que imprima de 1 até 100

```
Exemplo01.py x
1  for i in range(100):
2      print(i+1)
3
```

Exemplo 02 - Vamos criar um programa que vai mostrar somente os números ímpares de 1 a 100

```
Exemplo02.py x
1  for i in range(100):
2      print(i + 1)
3      if i % 2 == 1:
4          print(i)
5
```

Vamos fazer umas coisas legais agora, mas antes vamos falar de uma função em Python chamada `len()`.

Mas o que uma função???

Pense em uma função como uma pequena "fábrica" ou uma receita dentro do seu código. Ela é um bloco de comandos projetado para realizar uma tarefa específica.

Em vez de você escrever o mesmo código várias vezes, você o embala em uma função e o "chama" sempre que precisar.

Como funciona a estrutura?

Uma função geralmente segue três etapas simples:

Entrada (Parâmetros): O que a função recebe para trabalhar (ex: ingredientes).

Processamento: O que a função faz com esses dados (ex: o ato de cozinhar).

Saída (Retorno): O resultado final que ela entrega (ex: o prato pronto).

Por que usar funções?

Se você está começando a programar, as funções serão suas melhores amigas por três motivos principais:

Reutilização: Escreva uma vez, use mil vezes.

Organização: O código fica muito mais limpo e fácil de ler. Se algo der errado, você sabe exatamente em qual "caixinha" (função) o erro está.

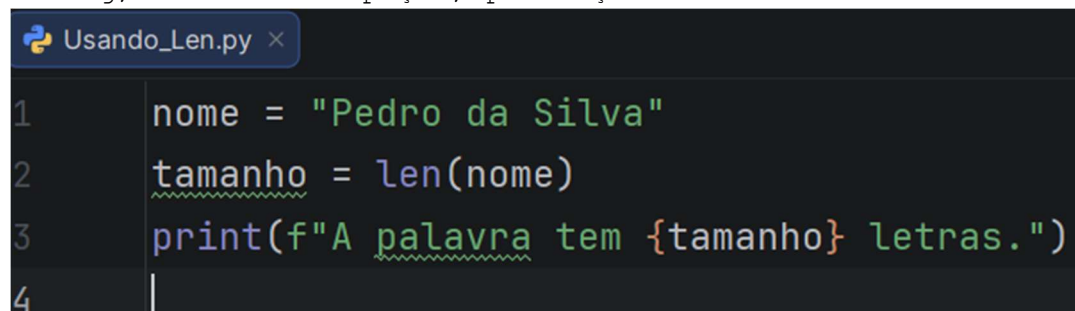
Abstração: Você não precisa saber como a função funciona internamente o tempo todo, apenas o que ela faz. (Você aperta o botão da cafeteira sem precisar entender a fiação interna dela).

O `len()` usamos para verificar o comprimento de uma string é extremamente simples e direto. A função `len()` (**abreviação de length, ou comprimento**).

Aqui estão as formas mais comuns de fazer isso:

1. Usando a função `len()`

Esta é a maneira padrão. Ela conta todos os caracteres dentro da string, incluindo espaços, pontuação e símbolos.



```
Usando_Len.py x
1 nome = "Pedro da Silva"
2 tamanho = len(nome)
3 print(f"A palavra tem {tamanho} letras.")
4
```

E agora se eu quisesse percorrer letra por letra desta palavra, como faríamos?

Modo 1 Forma direta → Já direto no arquivo que já estávamos trabalhando.

```

Usando_Len.py x
1 nome = "Pedro da Silva"
2 tamanho = len(nome)
3 print(f"A palavra tem {tamanho} letras.")
4
5 for letra in nome:
6     print(letra)

```

Modo 2 Usando o índice (Posição) →

Às vezes, você precisa saber a posição (índice) de cada letra (por exemplo: "A letra P está na posição 0"). Para isso, usamos `range()` e `len().letras`.

```

Usando_Len.py x
1 nome = "Pedro da Silva"
2 tamanho = len(nome)
3 print(f"A palavra tem {tamanho} letras.")
4
5
6 ## Forma Direta
7 for letra in nome:
8     print(letra)
9
10 ## Usando índice
11 for i in range(tamanho):
12     print(f"Índice {i}: {nome[i]}")

```

Vamos a um teste queremos pegar uma palavra/frase que será digitada pelo usuário e queremos que mostre as vogais e quantidade de vogais que existem nesta palavra/frase digitada.

```

Contando_Vogais.py x
1 palavra = input("Digite uma palavra: ")
2 tamanho = len(palavra)
3 vogais = "aeiouAEIOU"
4 contador = 0
5 for letra in palavra:
6     if letra in vogais:
7         print(f"Vogal encontrada {letra}")
8         contador = contador + 1
9
10 print(f"Foram encontrada {contador} vogais na palavra --> {palavra}")

```



Vamos fazer uma coisa legal, vamos pegar uma palavra/frase e fazer o seguinte vamos deixar esta palavra/frase toda em minúscula e depois vamos fazer somente as vogais em maiúscula. O programa dever a palavra/frase original e a alterada. Observe que vai acontecer, na verdade vai aparecer mais alguns "comandos" nas variáveis que são do

tipo string.

```
Muda_Vogal.py x
1  palavra_frase = input("Digite uma palavra/frase: ")
2  frase_minuscula = palavra_frase.lower();
3  print(frase_minuscula)
4  frase_final = ""
5  vogais = "aeiou"
6  for letra in frase_minuscula:
7      if letra in vogais:
8          frase_final += letra.upper()
9      else:
10         frase_final += letra
11 print(frase_final)
12
```

Um Desafio 01



Faça o mesmo programa anterior , só que agora coloque somente as consoantes em maiúsculas.

ATENÇÃO

Em Python, métodos são como "habilidades" ou ações que um objeto sabe realizar. Como tudo em Python é um objeto, uma string (texto) já vem de fábrica com várias funções internas que você pode usar para transformá-la ou analisá-la.

A sintaxe é sempre: `objeto.metodo()`.

Os Principais Métodos de String

Aqui estão os métodos que você mais usará no dia a dia para manipular textos:

1. Transformação de Caixa (Maiúsculas/Minúsculas)

Estes métodos alteram como as letras aparecem sem mudar o conteúdo da frase.

`.upper()`: Deixa tudo em MAIÚSCULO.

`.lower()`: Deixa tudo em minúsculo.

`.capitalize()`: Deixa apenas a primeira letra da frase em maiúscula.

`.title()`: Deixa a primeira letra de cada palavra em maiúscula.

```
1 frase = "Ola esta e uma frase de TESTE, esPeRo que tudo certo"
2 print(f"Usando upper() --> {frase.upper()}")
3 print(f"Usando lower() --> {frase.lower()}")
4 print(f"Usando capitalize() --> {frase.capitalize()}")
5 print(f"Usando title() --> {frase.title()}")
```

Resultado

```
C:\Python\Exercicios\.venv\Scripts\python.exe C:\Python\Exercicios\Transf_String.py
Usando upper() --> OLA ESTA E UMA FRASE DE TESTE, ESPERO QUE TUDO CERTO
Usando lower() --> ola esta e uma frase de teste, espero que tudo certo
Usando capitalize() --> Ola esta e uma frase de teste, espero que tudo certo
Usando title() --> Ola Esta E Uma Frase De Teste, Espero Que Tudo Certo
```

2. Limpeza e Espaçamento

Muito úteis para tratar dados digitados por usuários que podem conter espaços extras.

`.strip()`: Remove espaços em branco no início e no fim da string.

`.replace("antigo", "novo")`: Substitui uma parte do texto por outra.

```
LimpezaEspacamento_String.py x
1 frase = "  Ola vamo trabalhar com espaçamentos e limpeza  "
2 tamanho = len(frase)
3 print(f"Tamanho da frase sem usar strip(), {tamanho}")
4 frase_usando_strip = frase.strip()
5 tamanho = len(frase_usando_strip)
6 print(f"Tamanho da frase usando strip(), {tamanho}")
7 print()
8 print("##### Usando replace() #####")
9
10 frase = "Eu gosto de goiaba , goiaba é tudo de bom"
11 print(frase)
12 frase_usando_replace = frase.replace(old="goiaba ", new=" maçã ")
13 print(frase_usando_replace)
14
```

Resultado

```
C:\Python\Exercicios\.venv\Scripts\python.exe C:\Python\Exercicios\LimpezaEspacamento_String.py
Tamanho da frase sem usar strip(), 51
Tamanho da frase usando strip(), 45

##### Usando replace() #####
Eu gosto de goiaba , goiaba é tudo de bom
Eu gosto de maçã , maçã é tudo de bom
```

3. Pesquisa e Validação

Para saber o que existe dentro do seu texto.

```
.find("termo"): Retorna a posição (índice) onde a palavra começa.  
Se não achar, retorna -1.
```

```
.startswith("texto"): Verifica se a frase começa com aquele texto (retorna True ou False).
```

```
.endswith("texto"): Verifica se a frase termina com aquele texto.
```

.isdigit(): Verifica se a string contém apenas números.

[illegible]

4. Divisão e Junção

Essenciais para converter frases em listas e vice-versa.

.split(): Quebra a frase em uma lista de palavras (usa o espaço como padrão).

.join(lista): Junta os itens de uma lista para formar uma única string.

```
C:\Python\ExerciciosAvaliacao\.venv\Scripts\python.exe C:\Python\ExerciciosAvaliacao\Exercicio01.py
Digite uma frase: ola como vai
Frase Original --> OLA COMO VAI
Nova Frase --> oLa cOmO VaI
```

 DivisaoJuncao_String.py ×

```
1
2     titulo = "Como aprender Python em dez dias"
3     print(f"Título original: {titulo}")
4
5     # --- O MÉTODO .split() ---
6
7     lista_de_palavras = titulo.split()
8
9     print(f"Lista gerada: {lista_de_palavras}")
10
11    # --- O MÉTODO .join() ---
12
13    lista = ["evandrojv", "com", "br"]
14    link_final = ".".join(lista)
15
16    # Opcional: Deixar tudo minúsculo para o link ficar bonito
17    link_final = link_final.lower()
18
19    print(f"Link gerado: --> www.{link_final}")
```

```
C:\Python\Exercicios\.venv\Scripts\python.exe C:\Python\Exercicios\DivisaoJuncao_String.py
Título original: Como aprender Python em dez dias
Lista gerada: ['Como', 'aprender', 'Python', 'em', 'dez', 'dias']
Link gerado: --> www.evandrojv.com.br
```

Exercícios para avaliação

1 - Crie um programa onde será digitado uma frase qualquer, o programa deve tornar a frase toda em maiúscula e depois todas as letras que tiverem em posição par devem ficar em minúsculas. O programa deve retornar a frase original e a nova frase que foi gerada. Veja abaixo a imagem do programa rodando.

```
C:\Python\ExerciciosAvaliacao\.venv\Scripts\python.exe C:\Python\ExerciciosAvaliacao\Exercicio01.py
Digite uma frase: Ola como vai voce, tudo bem?
Frase Original --> OLA COMO VAI VOCE, TUDO BEM?
Nova Frase --> oLa cOmO VaI VoCe, TuDo bEm?
```

2- Crie um programa que mostre os números de múltiplos 3 e 5 ao mesmo de 01 a 100, deve começar de 1. Veja abaixo a imagem do programa rodando.

```
C:\Python\ExerciciosAvaliacao\.venv\Scripts\python.exe C:\Python\ExerciciosAvaliacao\Exercicio02.py
15
30
45
60
75
90
```

3 - Vamos começar com uma lista escrita assim
lista = ["Casa", "Barco", "Computador", "Empresa"], você continuar o programa, de modo que ele passe pela lista e imprima na tela apenas as palavras que possuem "sa". Dica para passar lista use
for i in range(len(lista)):

Veja imagem abaixo do programa funcionando.

```
C:\Python\ExerciciosAvaliacao\.venv\Scripts\python.exe C:\Python\ExerciciosAvaliacao\Exercicio03.py
['Casa', 'Barco', 'Computador', 'Empresa']
*****Encontrando palavras que contenham "sa" *****
casa
empresa
```

Programas Resolvidos

Desafio01

```
Muda_Consoante.py ×  
1  palavra_frase = input("Digite uma palavra/frase: ")  
2  frase_minuscula = palavra_frase.lower();  
3  print(frase_minuscula)  
4  frase_final = ""  
5  vogais = "aeiou"  
6  for letra in frase_minuscula:  
7      if not letra in vogais:  
8          frase_final += letra.upper()  
9      else:  
10         frase_final += letra  
11  print(frase_final)
```