
targettrust
treinamento e tecnologia

Prática Extra

Implemente um programa em Python que recebe como entrada duas *strings* digitadas pelo usuário (o programa só aceita *Strings* de mesmo tamanho), e como resultado imprime as duas strings alternadamente. Ou seja, deve-se imprimir o 1° char de str1, depois o 1° char de str2, e assim por diante.

Veja o exemplo:

str1 = "joão" str2 = "1234"
Saída = "j1o2a3o4"

10 min

Formatação de Strings

O método format nos auxilia na hora de formatar uma ou mais strings.

O uso mais comum é no comando print, quando queremos formatar a saída para um arquivo ou para a tela do usuário.

Exemplos:

- print('MEDIA = {:.5f}'.format(MEDIA))
- print("Essa é a aula de número {} do curso de Lógica com Python".format(4))
- usuario_cidade = "{} nasceu em {} e mora em {}"
- print(usuario_cidade.format("Matheus", "Alegrete", "Porto Alegre”))

Prática

Leia uma string grande (por exemplo o Hino Nacional Brasileiro). Após, responda as seguintes perguntas utilizando os métodos da string. Utilize o format em todos seus print.

a) Qual o tamanho da string?
b) Em qual posição da string aparece a primeira letra a?
b) Em qual posição da string aparece a última letra e?
d) Quantos espaços temos na string?

15 min

Comandos in/not in

Podemos verificar se um elemento está na lista ou não está utilizando esse operador

```
'Alegre' in 'Porto Alegre'
5 in [1, 2, 3, 4, 6]
5 not in [1, 2, 3, 4, 6]
```

targettrust
treinamento e tecnologia

Obrigado!

Erro no material? Envie e-mail para:
materiais@targettrust.com.br


targettrust
treinamento e tecnologia

Listas

Declaração de uma variável / constante

num1 = 5
num2 = 8
num3 = 6
num4 = 1

E se eu preciso armazenar 1000 números?

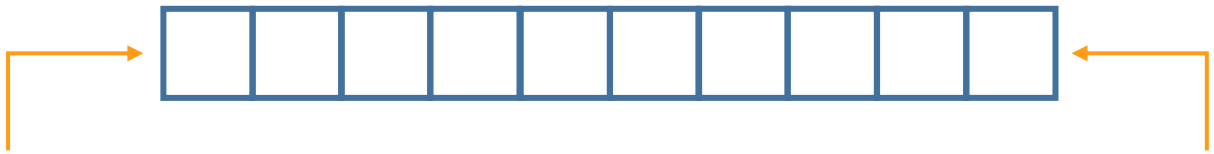
- Declaro 1000 variáveis?
- Não, usamos listas!

Declaração de listas

- num = []

Listas

Representação em 1 dimensão



Listas em Python são utilizadas para organizar e agrupar dados.
Os dados podem ser homogêneos ou heterogêneos!

- notas_prova_1 = [7.1, 8.5, 7.5, 4.0, 3.0]
- minha_lista = [5, 3.5, 'Matheus', True]

Listas - Slice

Também chamado de fatiamento

```
pessoas[2:5]
pessoas[3:7]
pessoas[:7]
pessoas[7:]
```

```
# Listas - Exemplos


meses = [Jan','Fev',’Mar','Abr','Mai','
Jun','Jul','Ago','Set','Out','Nov','Dez']

for mes in meses:
    print(mes)
```

```
# Listas - Exemplos

meses = [Jan','Fev',’Mar','Abr','Mai','
Jun','Jul','Ago','Set','Out','Nov','Dez']

print(meses[0])
print(meses[len(meses) - 1])

print(meses[-2])
print(meses[-1])
```

Prática

Implemente um programa em Python que recebe como entrada uma *string* digitada pelo usuário e retorna a *string* invertida.

Veja o exemplo:

Entrada = "existem várias maneiras de inverter uma string"
Saída = "gnirts amu retrevni ed sarienam sairáv metsixe"

10 min

Listas - Métodos

Mais utilizadas

- append, insert, remove, pop, index, count, sort, sorted, join, copy

Outras operações

- https://docs.python.org/pt-br/3.9/tutorial/datastructures.html

Dicionários

Tipo de dado que armazena mapeamentos chave / valor

```
nome_idade["Matheus"] = 25
nome_idade["João"] = 65
nome_idade["Maria"] = 12
```

Dicionários

Tipo de dado que armazena mapeamentos chave / valor

```
nome_idade["Matheus"] = 25
nome_idade["João"] = 65
nome_idade["Maria"] = 12
```

O que mais é possível fazer?

- alunos["Matheus"]["nota"] = 9.5
- alunos["Matheus"]["faltas"] = 4
- pedidos["Matheus"]["25/06/2020"].append("Mouse Cobra M711")
- pedidos["Matheus"]["25/06/2020"].append("Mousepad Taurus (930x300mm)")

Dicionários - Exemplos

```
siglas = {"TT" : "TargetTrust", "FB" :
"Facebook", "RS" : "Rio Grande do
Sul"}

print(siglas["TT"])
print(siglas["FB"])
print(siglas["RN"])
```

Retorna o elemento, se não existir, um erro ocorre.

```
notas = {"Bruna" : [8.3, 6.6], "Gabriel" : [0.9, 6.3], "Lucas" : [8
.0, 8.2]}

print(notas.get("Bruna", "Aluno não encontrado."))
print(notas.get("Lucas", "Aluno não encontrado."))
print(notas.get("Matheus", "Aluno não encontrado."))

notas["Matheus"] = [9.5, 8.3]

print(notas.get("Matheus", "Aluno não encontrado."))
```

```
notas = {"Bruna" : 7.4, "Gabriel" : 3.4, "Lucas" : 8.1}

print('chaves: {}\n'.format(notas.keys()))

print('for in keys')

for chave in notas.keys():
        print('\t' + chave)
print()
```

Percorre, ou seja, itera sobre as chaves do dicionário.

```
notas = {"Bruna" : 7.4, "Gabriel" : 3.4, "Lucas" : 8.1}

print('valores: {}\n'.format(notas.values()))

print('for in values')
for chave in notas.values():
        print('\t' + str(chave))
print()
```

Percorre, ou seja, itera sobre os valores de todas chaves do dicionário.

```
notas = {"Bruna" : 7.4, "Gabriel" : 3.4, "Lucas" : 8.1}

print('items: {}\n'.format(notas.items()))

print('for in items')
for chave, valor in notas.items():
  print('\t{} : {}'.format(chave, valor))
print()
```

Percorre, ou seja, itera sobre todos pares chave, valor

Sets

Para resolver determinados problemas precisamos rapidamente remover os elementos duplicados.

Exemplo:

```
produtos = ['Monitores', 'Monitores', 'Teclado', 'Mouse', 'Mousepad', 'Pen Drive', '
Pen Drive']
print(produtos)

produtos_sem_repeticao = set(produtos)
print(produtos_sem_repeticao)
```

Diversas funções e métodos para listas funcionam com sets.

Por exemplo: in, not in, add, pop, len

targettrust
treinamento e tecnologia

Obrigado!

Erro no material? Envie e-mail para:
materiais@targettrust.com.br