

**Título: *Strings* de dados - listas**

CENÁRIO DE APRENDIZAGEM	
<b>Escola:</b>	<b>Duração (minutos):</b> 45
<b>Professor(a):</b>	<b>Idade dos alunos:</b> 13

<b>Ideia Chave:</b>	Vamos encontrar <i>strings</i> de dados - listas
---------------------	--

**Tópicos:**

- Os alunos aprofundam sua compreensão do uso de vários softwares e políticas.

**Objetivos:**

- Os alunos serão capazes de projetar e criar programas que utilizam sub-rotinas, estruturas e tipos de dados apropriados, expressões, variáveis e comandos iterativos e condicionais.
- Linguagens de programação gerais são usadas para criar programas.
- Os alunos compreendem as diferentes maneiras de usar simulações e algoritmos de organização passo a passo para resolver problemas.

**Resultados:**

- Os alunos criam um jogo, aplicativo ou aplicativo móvel mais complexo que resolve um problema particular de um assunto ou tópico específico.
- Os alunos aprendem a delinear a operação de um programa mais complexo em vários padrões e generalizações.

**Formas de trabalho:**

- trabalho individual
- trabalho em pares
- trabalho de grupo

**Métodos:**

- apresentação
- discussão
- exercício interativo

## ARTICULAÇÃO

### Linha de atuação (duração, minutos)

### INTRODUÇÃO

O professor explica e inicia a discussão com os alunos:

Nos exemplos que vimos até agora, conhecemos diferentes tipos de dados input, com os quais voltaremos a trabalhar posteriormente. Se trabalhamos com múltiplos dados input, armazenamo-los em diferentes variáveis. Até mesmo quando os usamos para realizar a mesma operação, tivemos de os escrever segundo um comando especial, o que tornou as nossas soluções mais complexas. Podemos criar soluções mais simples?

### PARTE PRINCIPAL

Se precisarmos de escrever uma grande quantidade de dados, todos com uma característica comum, é aconselhável armazená-los numa estrutura única de dados. Assim, podemos usar os dados input através de nomes comuns e indexados, que nos dirão a posição de cada parte dos dados numa estrutura de dados maior. Alguns exemplos mais usuais são os nomes dos alunos, as temperaturas diárias do ar, e outros do mesmo género. A lista é, pois, uma estrutura de dados no Python que pode ser usada para armazenar vários itens numa única variável. Os itens de uma lista podem ser de qualquer tipo de dados, a que damos o nome de elementos da lista.

#### Exemplos de listas

<b>[8,5,3,7,6,2]</b>	Uma lista de números com 6 elementos
<b>['Ana','Ivo','Marko','Lucija','Nikola']</b>	Uma lista de cadeias de caracteres com 5 elementos
<b>[7,8,'Leona','Darko',3.14,1.4]</b>	Uma lista de 6 elementos - números inteiros e decimais e cadeias de caracteres
<b>[12,[2,5],[ ],['a','b'],'c']</b>	Uma lista com um número inteiro, uma lista com números, uma lista com caracteres e uma lista vazia

Os elementos de uma lista podem ser diferentes tipos de dados. Uma única lista pode conter números inteiros e decimais, caracteres e até mesmo outras listas.

Vamos usar alguns exemplos no nosso interface para mostrar diferentes formas de definir listas. As listas podem ser acrescentadas a uma variável, e também podemos fazer uma nova lista aplicando a adição (+) e a multiplicação (\*).

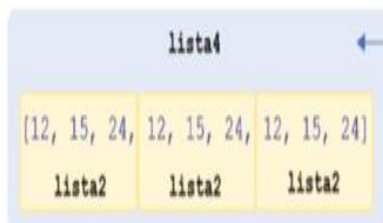
```
>>> lista1=[3,7,8,2,34,5]
>>> lista1
[3, 7, 8, 2, 34, 5]
>>> lista2=[12,15,24]
>>> lista2
[12, 15, 24]
```

```
>>> lista3=lista1+lista2
>>> lista3
[3, 7, 8, 2, 34, 5, 12, 15, 24]
>>>
```



Lista4 é o resultado da adição de duas listas (lista1 e lista2)

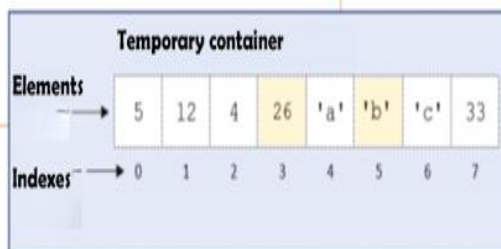
```
>>> lista4=lista2*3
>>> lista4
[12, 15, 24, 12, 15, 24, 12, 15, 24]
>>>
```



Lista4 é o resultado da multiplicação da lista2 com o número inteiro 3

### Uso de uma parte de uma lista com indexação

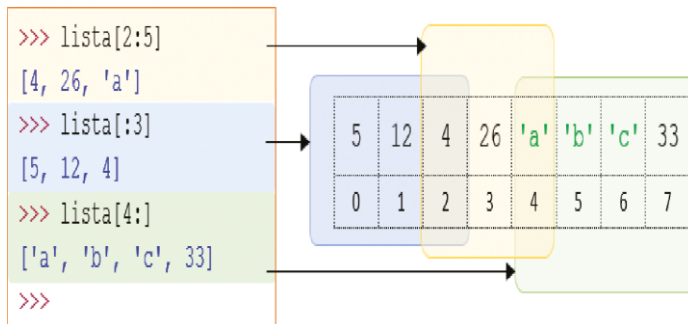
```
>>> lista=[5, 12, 4, 26, 'a', 'b', 'c', 33]
>>> print(lista)
[5, 12, 4, 26, 'a', 'b', 'c', 33]
>>> lista[3]
26
>>> lista[5]
'b'
>>>
```



Como podemos extrair diferentes elementos de uma lista? Para tal, usamos índices, da mesma forma que apanhámos elementos de uma string (cadeia, corda).

## Uma parte de uma lista

Tal como com as strings, uma parte de uma lista é definida digitando o primeiro e o último elemento que queremos, separados como dois pontos (:).



## EXERCÍCIO 1

Coloque todos os elementos de uma lista de dados única, um a seguir ao outro. Como podemos saber o número de elementos de uma lista? Tal como com as strings, usamos a função **len()**, que dá o resultado do número de elementos numa lista.

1. Definimos uma lista de nomes
2. Usamos **len()** para determinar o comprimento da lista
3. Usamos **loop** para publicar todos os elementos

```

>>> imena=['Ivan','Marko','Ana','Leona']
>>> for i in range (len(imena)):
    print(imena[i])

Ivan
Marko
Ana
Leona
>>>
    
```

1. definimos uma lista de nomes
2. utilizámos len(imena) para determinar o comprimento da lista
3. utilizámos um laço para imprimir todos os elementos

Número da repetição	Valor do índice no loop	Comando de execução	de	Ítems publicados
Número da repetição	Valor do índice no loop	Comando de execução		Ítems publicados
1. repetition	i=0	Print (imena[0])		Ivan
2. repetition	i=1	Print (imena[1])		Marko
3. repetition	i=2	Print (imena[2])		Ana
4. repetition	i=3	Print (imena[3])		Leona

## EXERCÍCIO 2

De acordo com o exemplo anterior, os alunos podem projetar, criar e testar seus próprios exemplos.

### CONCLUSÃO

Alunos e professor discutem e avaliam as soluções apresentadas.

#### **Métodos**

apresentação

discussão

trabalhar no texto

trabalho gráfico

exercício interativo / simulação no computador

entrevista

demonstração

representação

#### **Formas de trabalho**

Trabalho individual

Trabalho em pares

trabalho em equipa/grupo

trabalho frontal

#### **Material:**

- 

#### **Bibliografia**

- 

### OBSERVAÇÕES PESSOAIS, COMENTÁRIOS E NOTAS