



Programação de Aplicativos

Professor: Euclides Paim
euclidespaim@gmail.com



Listas em Python

Arrays

Professor: Euclides Paim
euclidespaim@gmail.com



Programação de Aplicativos

Listas em Python

Sumário

- Revisão
- *While Loops*
 - Sintaxe do *While*
- *For Loops*
 - Sintaxe do *For*
- Exercícios de fixação
- Resumo da aula





Programação de Aplicativos

Listas em Python

Listas

Listas são usadas para armazenar **múltiplos** itens em uma única variável.

As listas são um dos quatro tipos de dados integrados do Python usados para armazenar coleções de dados. Os outros três são:

- **Tupla,**
- **Conjunto (Set) e**
- **Dicionário (Dictionary)**

Todos com diferentes características e usos.



Programação de Aplicativos

Listas em Python

Exemplo:

Criando uma lista:

Sintaxe:

```
thislist = ["apple", "banana", "cherry"]  
print(thislist)
```



Programação de Aplicativos

Listas em Python

Itens

da

Lista

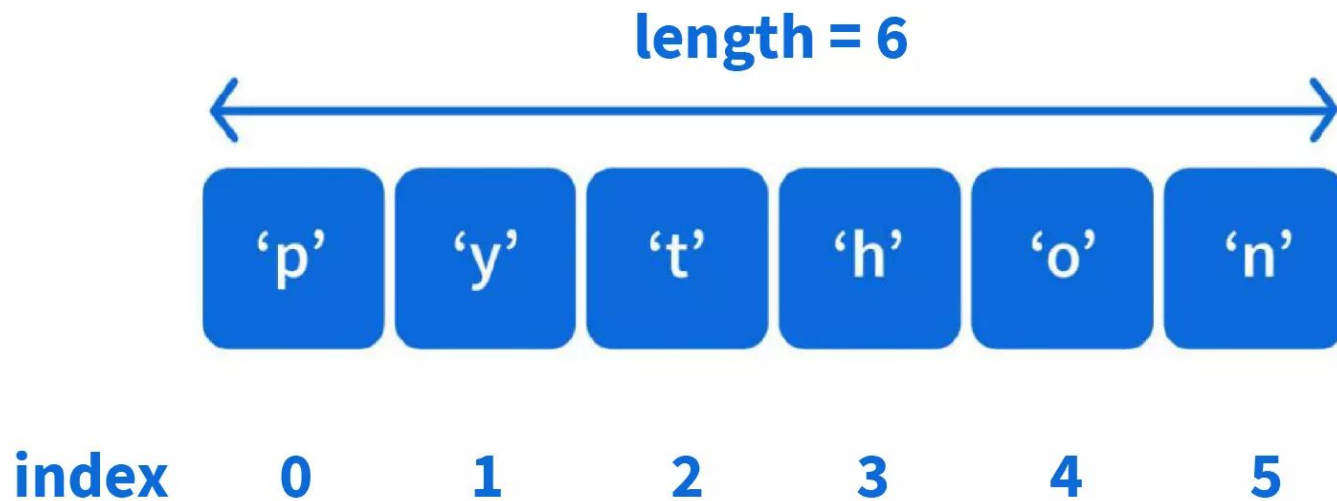
Os itens de uma lista são **ordenados**, **mutáveis** (podem ser modificados) e **permitem valores duplicados**.

Os itens da lista são **indexados**: o primeiro item tem índice `[0]`, o segundo item tem índice `[1]` e assim por diante.



Programação de Aplicativos

Listas em Python





Programação de Aplicativos

Listas em Python

Ordenado

Quando dizemos que as listas são ordenadas, isso significa que os itens possuem uma **ordem definida**, e essa ordem **não será alterada**.

Se você adicionar novos itens a uma lista, eles serão colocados **no final da lista**.

Nota: Existem alguns métodos de lista que podem **alterar a ordem**, mas, em geral, a **ordem dos itens permanece a mesma**.



Programação de Aplicativos

Listas em Python

Mutável

A lista é **mutável**, o que significa que podemos **modificar**, **adicionar** e **remover** itens em uma lista **após ela ter sido criada**.



Programação de Aplicativos

Listas em Python

Permite Duplicatas

Como as listas são **indexadas**, elas podem conter **itens com o mesmo valor**:

Exemplo:

```
thislist = ["apple", "banana", "cherry", "apple", "cherry"]  
print(thislist)
```



Programação de Aplicativos

Listas em Python

Tamanho da Lista

Para determinar quantos itens uma lista possui, use a função `len()`:

Exemplo:

```
thislist = ["apple", "banana", "cherry"]  
print(len(thislist))
```



Programação de Aplicativos

Listas em Python

Adicionar Itens

Para adicionar um item ao final da lista, use o método `append()`:

Exemplo

Usando o método `append()` para adicionar um item:

```
lista = ["apple", "banana", "cherry"]  
lista.append("orange")  
print(lista)
```



Programação de Aplicativos

Listas em Python

Inserir Itens

Para inserir um item da lista em um índice específico, use o método `insert()`.

O método `insert()` insere um item na posição (índice) especificada:

Exemplo

Inserir um item na **segunda posição**:

```
thislist = ["apple", "banana", "cherry"]  
thislist.insert(1, "orange")  
print(thislist)
```



Programação de Aplicativos

Listas em Python

Remover um Item Específico

O método `remove()` remove o **item especificado** da lista.

Exemplo

Remover **"banana"** na **segunda posição**:

```
thislist = ["apple", "banana", "cherry"]  
thislist.remove("banana")  
print(thislist)
```

Nota: Se houver **mais de um item com o valor especificado**, o método `remove()` irá remover **apenas a primeira ocorrência**.



Programação de Aplicativos

Listas em Python

Remover um Índice Específico

O método `pop()` remove o item que está no **índice especificado**.

Exemplo

Remover o **segundo item**:

```
thislist = ["apple", "banana", "cherry"]  
thislist.pop(1)  
print(thislist)
```

Nota: Se você **não especificar o índice**, o método `pop()` irá remover o **último item da lista**.



Programação de Aplicativos

Listas em Python

Resumo: **list** em Python

- Estrutura que armazena **vários valores** em uma única variável.
- Os elementos são **ordenados e indexados** (começam no índice **0**).
- Podem conter qualquer tipo de dado: números, textos, etc.



Programação de Aplicativos

Listas em Python

Resumo: **list** em Python

Principais operações:

- **Criar:** `nomes = ["Ana", "Carlos", "Maria"]`
- **Acessar:** `nomes[0]` → Ana
- **Adicionar:** `nomes.append("João")`
- **Remover:** `nomes.remove("Carlos")`
- **Tamanho:** `len(nomes)`
- **Percorrer:**

```
for nome in nomes:  
    print(nome)
```



That's all Folks!



Programação de Aplicativos

Sintaxe, Variáveis e Tipos de Dados

Exercícios:

Exercício 1: Contador simples (nível fácil)

Enunciado:

Escreva um programa que use um laço `while` para exibir os números de 1 a 10 na tela.

Exemplo de saída esperada:

```
1
2
3
...
10
```



Programação de Aplicativos

Sintaxe, Variáveis e Tipos de Dados

Exercícios:

Exercício 2: Peça para o usuário digitar números inteiros. O programa deve continuar pedindo números até que o usuário digite 0. Ao final, mostre a soma de todos os números digitados (exceto o zero).

Exemplo:

```
Digite um número: 5
Digite um número: -2
Digite um número: 3
Digite um número: 0
Soma total: 6
```



Programação de Aplicativos

Sintaxe, Variáveis e Tipos de Dados

Exercícios:

Exercício 3: Implemente um programa que solicite a senha de acesso ao usuário. O laço `while` deve continuar pedindo a senha até que o usuário acerte. A senha correta é `"python123"`.

Extra: Mostre uma mensagem de sucesso ao final.

Exemplo:

```
Digite a senha: admin
Senha incorreta.
Digite a senha: 1234
Senha incorreta.
Digite a senha: python123
Acesso concedido!
```



Referências

Referências Básicas

CORMEN, Thomas H.; LEISERSON, Charles E.; RIVEST, Ronald L.; STEIN, Clifford. ***Algoritmos: teoria e prática***. 3. ed. Rio de Janeiro: Elsevier, 2013.

SEBESTA, Robert W. ***Conceitos de linguagens de programação***. 10. ed. São Paulo: Pearson, 2018.

MANZANO, José Augusto N. G.; OLIVEIRA, Jayr Figueiredo de. ***Algoritmos: lógica para desenvolvimento de programação de computadores***. 27. ed. São Paulo: Érica, 2016.

DOWNEY, Allen B. ***Pense em Python: como pensar como um cientista da computação***. Tradução de Cássio de Souza Costa. 2. ed. São Paulo: Novatec, 2016.

Referências Complementares

IEPSEN, Edécio Fernando. ***Lógica de programação e algoritmos com JavaScript***. 2. ed. São Paulo: Novatec, 2022.

Referências na Internet

<https://www.w3schools.com/python/>