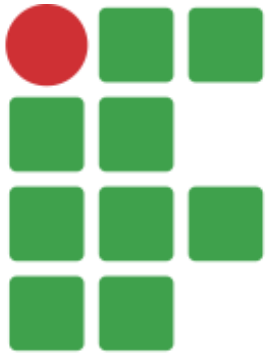


INSTITUTO FEDERAL
Paraíba

Campus Cajazeiras

ALGORITMOS E LÓGICA DE PROGRAMAÇÃO

PROFESSORES LEANDRO, MICHEL
CAJAZEIRAS / IFPB

Listas

OBJETIVOS

- Aprender sobre os estruturas de dados em Python;
- Aprender como usar dados em sequência no Python;
- Estudar a estrutura Lista;

ROTEIRO

- Estruturas em sequência
- Listas e matrizes

Estruturas em sequência

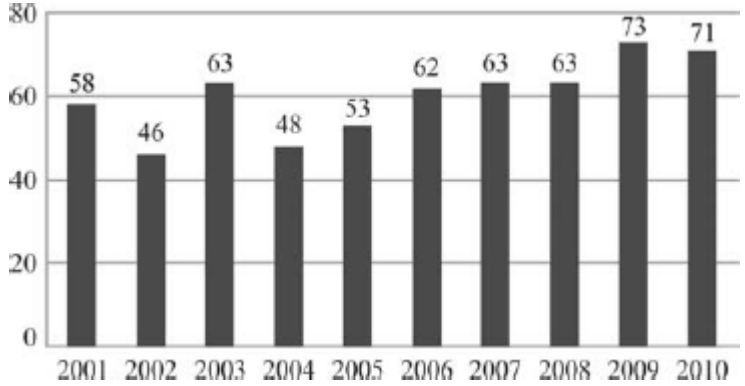
- Sequências!
  - A **sequência** numérica nada mais é do que uma **sequência** de números.
  - Podem ser **indexados** por algum valor **ordinal posicional**.
  - Uma sequência pode ser finita ou infinita.
  - Uma sequência pode ser crescente, descrente, constante ou oscilante.
- Lista: lista1 = [1, -2, 3, "IFPB", "ifpb", "ALGORITMOS"]
- Tupla: tupla1 = (1, "IFPB', 3.14156, (2,3), "algoritmos")
- String: oracao1 = "vamos para a aula de algoritmos!"

https://youtu.be/dX5SL2YbjWw

Estruturas em sequência

- Porque as sequências são importantes?

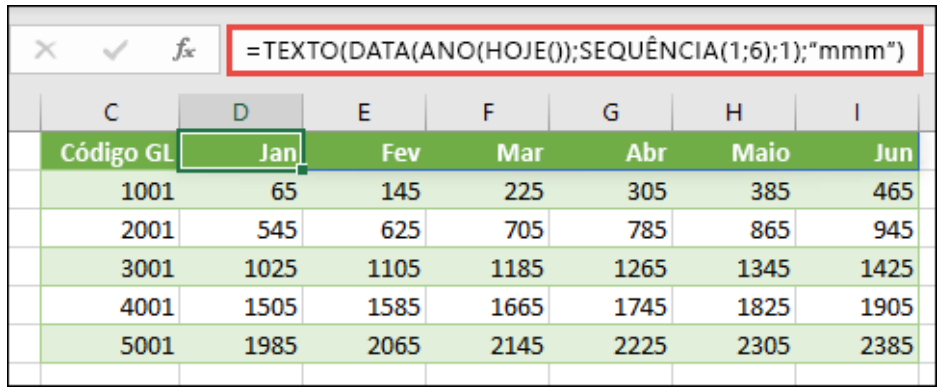
https://www.clubedopoker.com/img/regrasdepoker_obaralhodepoker_clip_image002.jpg



https://enem.estuda.com/questoes/

1.(2, 4, 6, 8, 10, 12, ...) sequência de números pares positivos;
2.(2, 3, 5, 7, 11, 13...) sequência de números primos;
3.(I, II, III, IV, V, VI, VI...) sequência de números romanos;
4.(2, 4, 6, 8, 10, 12) sequência de números positivos, pares e menores que 14.

https://www.guiaestudo.com.br/sequencia-numerica



Código GL	Jan	Fev	Mar	Abr	Maio	Jun
1001	65	145	225	305	385	465
2001	545	625	705	785	865	945
3001	1025	1105	1185	1265	1345	1425
4001	1505	1585	1665	1745	1825	1905
5001	1985	2065	2145	2225	2305	2385

https://support.content.office.net/pt-br/media/67d36c71-2823-4342-968c-e0a24aff2b2b.png

Listas

Sintaxe

lista = [a, b, ..., z]

Listas são **coleções heterogêneas (**diversidade**) de objetos.**

- **As listas estão ordenadas** - as listas lembram a ordem dos itens inseridos.
- **Acessado por índice** - os itens em uma lista podem ser acessados por meio de um índice.
- **As listas podem conter qualquer tipo de objeto** - podem ser números , strings , tuplas e até outras **listas**.
- **As listas são alteráveis (mutáveis)** - você pode alterar uma lista no local, adicionar novos itens e excluir ou atualizar itens existentes.

Listas

Exemplos

Existem várias maneiras de criar uma nova lista; o mais simples é colocar os valores entre colchetes[]

Os itens de uma lista **não precisam ser do mesmo tipo**. A lista a seguir contém um inteiro, uma string, um float, um número complexo e um booleano.

main.py

```
1   # Uma lista de inteiros
2   L1 = [1, 2, 3]
3
4   # Uma lista de strings
5   L2 = ["vernelho", "verde", "azul"]
6
7   # Uma lista de tipos de dados mistos
8   L3 = [ 1, "abc", 1.23, (3+4j), True]
9
10  print(L1)
11  print(L2)
12  print(L3)
13  # Imprime [1, 2, 3]
14  # Imprime ['vernelho', 'verde', 'azul']
15  # Imprime [1, 'abc', 1.23, (3+4j), True]
```

```
[1, 2, 3]
['vernelho', 'verde', 'azul']
[1, 'abc', 1.23, (3+4j), True]
>
```

Listas

Exemplos

O construtor list ()

Você pode converter outros tipos de dados em listas usando o construtor **list()** do Python

main.py

```
1  # Converta uma string em uma lista
2  L1 = list('abc')
3  print(L1)
4  # Imprime ['a', 'b', 'c']
5
6  # Converta uma tupla em uma lista
7  L2 = list((1, 2, 3))
8  print(L2)
9  # Imprime [1, 2, 3]
```

```
['a', 'b', 'c']
[1, 2, 3]
>
```

Listas

Exemplos

Indexação de lista

-5	-4	-3	-2	-1
Vermelho	Verde	Azul	Amarelo	preto
0	1	2	3	4

main.py

```
1  L = ["vermelho", "verde", "azul", "amarelo", "preto"]
2  print(L[2])
3  # imprime azul
4  print(L[-1])
5  # Imprima preto
6  print(L[-2])
7  # Imprima amarelo
```

```
azul
preto
amarelo
```

Listas

Exemplos

Alterar o valor do item

Você pode substituir um elemento existente por um novo valor atribuindo o novo valor ao índice.

-3	-2	-1
Vermelho	Verde	Azul
0	1	2

main.py

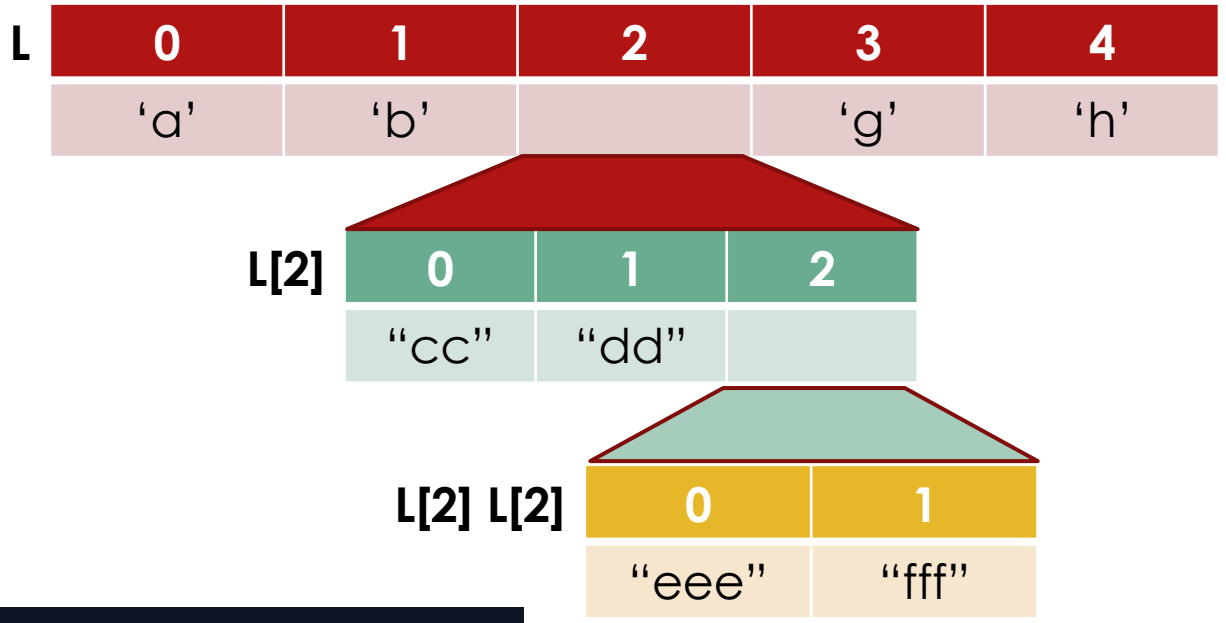
```
1  L = ["vermelho", "verde", "azul"]
2
3  L[0] = "amarelo"
4  print(L)
5  # Imprime ['amarelo', 'verde', 'azul']
6
7  L[-1] = "violeta"
8  print(L)
9  # Impressões ['amarelo', 'verde', 'violeta']
```

```
['amarelo', 'verde', 'azul']
['amarelo', 'verde', 'violeta']
>
```

Listas

Exemplos

Acessar itens da lista aninhada



main.py

```
1  L = ['a', 'b', ["cc", "dd", ["eee", "fff"]], 'g', 'h']
2  print(L[2][2])
3  # imprima ['eee', 'fff']
4  print(L[2][2][0])
5  # Imprima eee
```

```
['eee', 'fff']
eee
>
```

Listas

Se desejamos inserir um item em uma posição específica em uma lista, use o método **insert()** . Observe que todos os valores na lista após o valor inserido serão movidos um índice.

Exemplos

Adicionar itens a uma lista

Para adicionar novos valores a uma lista, use o método **append()**. Este método adiciona itens apenas ao final da lista.

main.py

```
1  L = ["vermelho", "verde", "amarelo"]
2  L.append("azul")
3  print(L)
4  # Imprime ['vermelho', 'verde', 'amarelo', 'azul']
5
```

```
['vermelho', 'verde', 'amarelo', 'azul']
>
```

main.py

```
1  L = ["vermelho", "verde", "amarelo"]
2  L.insert(1,"azul")
3  print(L)
4  # Imprime ['vermelho', 'azul', 'verde', 'amarelo']
5
```

```
['vermelho', 'azul', 'verde', 'amarelo']
>
```

Listas

Operações em Listas

A tabela abaixo mostra um resumo dos principais operadores para manipulação de listas em Python.

Operador	Descrição	Exemplo
+	concatena duas listas	[1, 2, 3] + [4, 5] retorna [1, 2, 3, 4, 5]
*	repete (replica) uma lista múltiplas vezes	[0, 1] * 2 retorna [0, 1, 0, 1]
[i]	retorna o i-ésimo elemento da lista	[10, 20, 30] print[1] retorna 20
[i:j]	retorna a sub-lista que vai dos índices i até j-1	[1, 3, 5, 7, 9] print[1:3] retorna [3, 5]
a = b[:]	2 cópias independentes	Exemplos Modificando listas
a = b	os 2 referenciam o mesmo objeto	

Listas

métodos em Listas

A tabela abaixo mostra um resumo dos principais métodos para manipulação de listas em Python.

método	Descrição	Exemplo
sort	Ordenando listas	lista.sort()
append	Adiciona elementos no final da lista.	lista.append('A')
extend	Une as duas lista, igual ao operador +	Lista.extend(['B', 'C'])
insert	inserir elementos em posições específicas da lista	lista.insert(3, 'D')
count	contar quantas vezes um elemento aparece numa lista	lista.count('D')
remove	remove elementos	lista.remove('B')
reverse	inverte a ordem da lista	list.reverse()

Listas

funções em Listas

A tabela abaixo mostra um resumo das principais funções para manipulação de listas em Python.

método	Descrição	Exemplo
sum	Soma todos os elementos	Soma = sum(lista)
len	Descobrir o tamanho de uma lista	tamanho = len(lista)
Sorted	Retornar essa cópia ordenada da lista	listaOrdenada = sorted(lista)
max	Retorna o maior item	Maior = max(lista)
min	Retorna o menor item	Menor = min(lista)

Exercícios 1

1 – ler uma lista de 5 número inteiros e imprimir cada número juntamente com a sua posição na lista.

2 – ler uma lista de 5 número reais e imprimir a lista na ordem inversa.

3 – ler uma lista de 4 notas e em seguida mostra as notas e a média.

4 – ler uma lista de 5 números e imprimir o menor e maior valor.