

Lista 1

Imagine uma situação na qual precisamos escrever um programa que armazena uma lista de frutas comercializadas em uma loja de hortifrúti, se o objetivo for pura e simplesmente anotar os nomes dessas frutas, o trabalho pode ser realizado utilizando uma string, no entanto, a linguagem python possui uma estrutura nativa que permite organizar coleções de maneira simples e muito efetiva, esse tipo de estrutura é chamado de lista (ou array).

Para criar uma lista em python, podemos utilizar a seguinte sintaxe:

```
frutas = ["Banana", "Maçã", "Uva", "Kiwi"]
```

Utilizando colchetes, podemos indicar que o valor que está sendo atribuído a uma variável é na verdade uma lista de valores, sendo que cada elemento da lista pode ser de um tipo diferente, veja os exemplos:

```
numeros_inteiros = [1, 2, 3]
numeros_fracionarios = [1.1, 2.2, 3.3]
misto = ["Chave", 2, 6.78]
```

Todas as listas são ordenadas, de forma que cada item da lista possui um índice único, que permite identificar a posição daquele valor dentro da lista. Listas no geral são estruturas bastante liberais, e permitem inserção de elementos duplicados, no entanto, cada elemento continua a ter um índice único na lista.

De maneira semelhante a strings, podemos acessar elementos dentro de uma lista a partir do seu índice.

```
frutas = ["Banana", "Maçã", "Uva"]
print(frutas[0]) # "Banana"
```

Também podemos acessar elementos de trás para frente utilizando valores negativos, de forma que o -1 é o ultimo item da lista, -2 o penúltimo e assim sucessivamente:

```
frutas = ["Banana", "Maçã", "Uva"]
print(frutas[-1]) # "Uva"
print(frutas[-2]) # "Maçã"
```

Outra funcionalidade interessante, é o acesso a uma sublista de itens utilizando um índice inicial (inclusivo) e outro final (não inclusivo), vamos a um exemplo:

```
frutas = ["Banana", "Maçã", "Uva", "Kiwi", "Laranja", "Amora", "Cereja"]
print(frutas[3:6]) # ['Kiwi', 'Laranja', 'Amora']
```

Pode-se utilizar valores negativos na seleção de sublista, e também podemos deixar o índice inicial ou final como vazios, dessa forma, será criada uma sublista a partir do ponto de início, ou até o ponto de fim.

```
frutas = ["Banana", "Maçã", "Uva", "Kiwi", "Laranja", "Amora", "Cereja"]
print(frutas[3:]) # ['Kiwi', 'Laranja', 'Amora', 'Cereja']
print(frutas[:4]) # ['Banana', 'Maçã', 'Uva']
```

Se precisar alterar algum dos valores de qualquer índice da lista basta atribuir um novo valor a ele:

```
frutas = ["Banana", "Maçã", "Uva", "Kiwi", "Laranja", "Amora", "Cereja"]
frutas[1] = "Carambola"
print(frutas) # ["Banana", "Carambola", "Uva", "Kiwi", "Laranja", "Amora", "Cereja"]
```

Para trocar vários valores de uma só vez, basta fornecer os índices em formato de sublista:

```
frutas = ["Banana", "Maçã", "Uva", "Kiwi", "Laranja", "Amora", "Cereja"]
frutas[2:4] = ["Carambola", "Tomate"]
print(frutas) # ['Banana', 'Maçã', 'Carambola', 'Tomate', 'Acerola', 'Kiwi', 'Laranja', 'Amora', 'Cereja']
```

Se for fornecida uma sublista com quantidade de itens maior do que especificado, estes serão inseridos a lista, e o restante dos itens são movimentados de acordo.

```
frutas = ["Banana", "Kiwi", "Laranja", "Amora", "Cereja"]
frutas[2:3] = ["Carambola", "Tomate", "Acerola"]
print(frutas) # ['Banana', 'Kiwi', 'Carambola', 'Tomate', 'Acerola', 'Amora', 'Cereja']
```

Podemos utilizar o if para verificar se a lista contém um item específico:

```
frutas = ["Banana", "Kiwi", "Laranja", "Amora", "Cereja"]
if "Banana" in frutas:
    print("Temos bananas")
```