**CEUB - FATECS**

**Lógica de Programação - Prof. Barbosa**

**Python - *list* (lista)**

**A lista também é conhecida com vetor ou array.**

**Lista é um conjunto sequencial de valores mutáveis, onde cada valor é identificado através de um índice (posição).**

**O primeiro valor armazenado na lista tem índice 0 (posição 0).**

**numeros = [10, 20, 30, 40, 50, 60] <- Conteúdo ou dado da lista**

**0 1 2 3 4 5 <- Índice ou posição da lista**

**nomes = [‘Alice’, ‘Samuel’, ‘Emily’, ‘Sofia’, ‘Vinícius’] <- Conteúdo ou dado da lista**

**0 1 2 3 4 <- Índice ou posição da lista**

**- Diferença entre variável e lista:**

**a. Variável armazena apenas um valor.**

**b. Lista armazena vários valores e são identificados pelo índice (posição).**

**Uma lista em Python é uma estrutura heterogênea e pode ter valores de qualquer tipo, incluindo outra lista ou uma tupla.**

**Exemplo:**

**lista1 = [3, 'abacate', 9.7, [5, 6, 3], "Python", (3, 'j')] 🡨 Conteúdo ou dados na lista**

**0 1 2 3 4 5 🡨 Índice ou posição da lista**

**print(lista1[2])**

**9.7**

**print(lista1[3])**

**[5, 6, 3]**

**print(lista1[3][1])**

**6**

**1. Uma lista em Python é declarada da seguinte forma.**

**1.1. Cria lista vazia, duas formas válidas:**

**lista\_um = [ ]**

**lista\_um = list() # Construtor list**

**1.2. Cria lista não vazia:**

**0 1 2 3 🡨 Índice ou posição da lista, positivo.**

**lista\_dois = [10, 20, 30, 40]**

**-4 -3 -2 -1 🡨 Índice ou posição da lista, negativo.**

**Obs.:**

**O primeiro valor da lista está no índice 0 (zero).**

**O último valor da lista está no índice -1.**

**Exemplo 1:**

**print(lista\_dois[1])**

**20**

**Exemplo 2:**

**print(lista\_dois[-3])**

**20**

**- Manutenção de uma lista ou CRUD (Create, Read, Update e Delete)**

**2. Insere (create)**

**3. Consulta (read)**

**4. Atualiza ou substitui (update)**

**5. Remove (delete)**

**2. Insere (create)**

**Sintaxe: lista.append(*x*)**

**Adiciona o elemento x no final da lista.**

**Ex.: antes da utilização do método:**

**lista = [1, 2, 3, 4, 5]**

**depois:**

**lista.append(9)**

**print(lista)**

**[1, 2, 3, 4, 5, 9]**

**Sintaxe; lista.insert(*i*, *x*)**

**Insere o item ‘x’ na posição ou índice ‘i’ e desloca os valores para direita.**

**O primeiro argumento é o índice do elemento (i) antes do qual será feita a inserção.**

**Ex.: antes da utilização do método:**

**lista = [1, 2, 3, 4, 5]**

**depois:**

**lista.insert(2, 6)**

**print(lista)**

**[1, 2, 6, 3 ,4, 5]**

**Obs.: se a posição não existir, o valor será inserido no final da lista.**

**lista = [1, 2, 3, 4, 5]**

**lista.insert(22, 6)**

**print(lista)**

**[1, 2, 3, 4, 5, 6]**

**3. Consulta (read)**

**Sintaxe 1: print(lista)**

**Consulta e mostra todos os valores armazenados na lista na horizontal.**

**Sintaxe 2: for item in lista:**

**Consulta e mostra todos os valores armazenados na lista na vertical,**

**for item in lista:**

**print(item)**

**Sintaxe 3: x in lista:**

**O in verifica se o valor ’x’ pertence à lista. Retorna *True* ou *False***

**Ex. 1:**

**lista = [10, 20, 30, 40]**

**if 30 in lista:**

**print('Valor está na lista')**

**else:**

**print('Valor não está na lista')**

**Ex. 2:**

**if 80 in lista:**

**print('Valor está na lista')**

**else:**

**print('Valor não está na lista')**

**Sintaxe 4: lista.index( x[, *start* [, *end*]] )**

**. Retorna o índice do primeiro item cujo valor é igual a ‘x’.**

**. Os argumentos opcionais *start* (início) e *end* (fim) são interpretados como na *slice notation* e são usados ​​para limitar a pesquisa a uma subsequência específica da lista.**

**. O índice retornado é calculado em relação ao início da sequência completa em vez do argumento inicial.**

**. Gera o ValueError se não existir o item ‘x’.**

**Ex.: lista = [10, 20, 30, 40, 50]**

**print(lista.index(30))**

**2**

**Obs.: o primeiro índice (posição) da lista é zero.**

**4. Atualiza ou substitui (update)**

**lista1 = [3, 'abacate', 9.7, [5, 6, 3], "Python", (3, 'j')]**

**Sintaxe:**

**lista [i] = ‘novo\_valor’**

**Usando notação de vetor.**

**O valor da posição ‘i’ será substituído pelo novo\_valor.**

**Exemplo 1:**

**lista1 [3] = 'morango'**

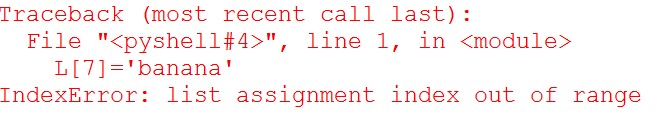
**print(lista1)**

**[3, 'abacate', 9.7, ‘morango’, "Python" , (3 , 'j')]**

**Exemplo 2:**

**lista1 [7] = 'banana'**

**A tentativa de acesso a um índice inexistente resulta em erro.**

****

**5. Remove (delete)**

**Sintaxe: lista.remove(*x*)**

**Remove o primeiro item encontrado na lista com valor = ‘x’ e desloca os valores para esquerda.**

**Se não existir o valor x, uma exceção ValueError é levantada.**

**Ex.: antes da utilização do método:**

**lista = [10, 20, 30, 40, 50]**

**depois:**

**lista.remove(40)**

**print(lista)**

**[10, 20, 30, 50]**

**6. Ordena**

**Sintaxe: lista.sort( [*reverse=False]* )**

**Ordena os elementos da lista in-place, ou seja, modifica a lista original no mesmo lugar na memória.**

**Coloca os elementos da lista em uma ordem crescente e altera a lista original.**

**Altera a lista original e retorna *None*.**

**reverse: se definido como False, ordena em ordem crescente.**

**reverse: se definido como True, ordena em ordem decrescente.**

**Ex.: antes da utilização do método:**

**lista = [4, 1, 5, 2, 3]**

**depois:**

**lista.sort()**

**print(lista)**

**[1, 2, 3, 4, 5]**

**Sintaxe: sorted(lista)**

**Retorna os elementos da lista em uma ordem crescente e não altera a lista original:**

**Ex.: antes da utilização do método:**

**lista = [3, 4, 1, 5, 2]**

**depois:**

**nova\_lista = sorted(lista)**

**print(nova\_lista)**

**[1, 2, 3, 4, 5]**

**print(lista)**

**[3, 4, 1, 5, 2]**

**Sintaxe: lista.reverse()**

**Inverta os elementos da lista, não coloca os dados em ordem decrescente.**

**Altera a lista original e retorna *None*.**

**Ex.: antes da utilização do método:**

**lista = [1, 2, 7, 3, 4, 5]**

**depois:**

**lista.reverse()**

**print(lista)**

**[5, 4, 3, 7, 2, 1]**

**Sintaxe: lista1 = list(reversed(lista)) # builtins**

**# Não altera a lista original, retorna o resultado na nova lista**

**Exemplo 1:**

**lista = [1, 2, 3, 4, 5]**

**nova\_lista = list(reversed(lista)) # Não altera a lista original, retorna o resultado na nova lista.**

**print(nova\_lista)**

**[5, 4, 3, 2, 1]**

**print(lista)**

**[1, 2, 3, 4, 5]**

**7. Numérico**

**Sintaxe: max(lista)**

**Retorna o maior elemento da lista:**

**Ex.: antes da utilização do método:**

**lista = [1, 2, 3, 4, 5]**

**depois:**

**m = max(lista)**

**print(m)**

**5**

**Sintaxe: min(lista)**

**Retorna o menor elemento da lista:**

**Ex.: antes da utilização do método:**

**lista = [1, 2, 3, 4, 5]**

**depois:**

**m = min(lista)**

**print(m)**

**1**

**Sintaxe: sum(lista)**

**Retorna a soma dos elementos da lista:**

**Ex.:**

**antes da utilização do método:**

**lista = [1, 2, 3, 4, 5]**

**depois:**

**soma = sum(lista)**

**print(soma)**

**15**

**Sintaxe: len(lista)**

**Retorna a quantidade de elementos da lista.**

**Ex.: antes da utilização do método:**

**lista = [1, 2, 3, 4, 5]**

**depois:**

**qtd = len(lista)**

**print(qtd) # print(len(lista))**

**5**

**Sintaxe: lista.count(*x*)**

**Retorna o número de vezes que o item ‘x’ aparece na lista.**

**Ex. 1:**

**lista = [1, 2, 2, 2, 3, 4, 5]**

**print(lista.count(2))**

**3**

**Ex. 2:**

**print(lista.count(9))**

**0**

## **- Funções para manipulação de listas**

**A lista é uma estrutura mutável, ou seja, ela pode ser modificada.**

**Tabela - Principais operações com listas**

|  |  |  |
| --- | --- | --- |
| **Função** | **Descrição** | **Exemplo** |
| **len** | **retorna o tamanho da lista.** | **li = [1, 2, 3, 4] len(li):**  **4** |
| **count** | **retorna quantas vezes x aparece na lista** | **li = [1, 2, 3, 3, 4]**  **count(3):**  **2** |
| **min** | **retorna o menor valor da lista.** | **li = [10, 40, 30, 20] min(li):**  **10** |
| **max** | **retorna o maior valor da lista.** | **li = [10, 40, 30, 20] max(li):**  **40** |
| **sum** | **retorna soma dos elementos da lista.** | **li = [10, 20, 30] sum(li):**  **60** |
| **append** | **adiciona um novo valor na no final da lista.** | **li = [1, 2, 3]**  **li.append(100)**  **li: [1, 2, 3, 100]** |
| **in** | **verifica se um valor pertence à lista.** | **li = [1, 2 , 3, 4]**  **3 in li:**  **True** |
| **sort()** | **ordena em ordem crescente** | **li = [3, 5, 2, 4, 1, 0]**  **li.sort()**  **li: [0, 1, 2, 3, 4, 5]** |

**Exemplo:**

**l1 = [3, 5, 2, 4, 1, 0]**

**l1.reverse()**

**[0, 1, 4, 2, 5, 3]**

**- Obs.: o reserse não coloca os dados em ordem decrescente.**

**Prof. Barbosa.**