

CAPÍTULO 5

Listas

Listas são estruturas de dados muito versáteis e úteis. Elas são usadas para armazenar coleções ORDENADAS de elementos, onde cada elemento pode ser de qualquer tipo de dado, como números, strings, booleanos e até mesmo outras listas e estruturas de dados. As listas são mutáveis, o que significa que seus elementos podem ser alterados, adicionados ou removidos após a criação da lista.

Para criar uma lista você pode utilizar colchetes `[]` e separar os elementos por vírgulas. Também é possível criar uma lista vazia e adicionar elementos posteriormente. Para isso, utilize a função `list()` ou simplesmente colchetes vazios. Por exemplo:

```
1 >>> lista_vazia = list()
2 >>> lista_vazia = [] #outra alternativa
3 >>> bases_dna = ["A", "C", "G", "T"]
4 >>> bases_rna = ["A", "C", "G", "U"]
5 >>> numeros = [0, 1, 2, 3]
6 >>> genes = ["APOE", "TOMM40", "FTO"]
```

As listas são indexadas, o que significa que cada elemento possui uma posição única dentro da lista (assim como nas *strings*). Os índices começam em 0 para o primeiro elemento e vão até (tamanho da lista - 1). Você pode acessar elementos

individuais de uma lista usando o operador de indexação (`[]`). Por exemplo:

```
1 >>> numeros = [1, 2, 3, 4, 5]
2 >>> primeiro_elemento = numeros[0] # Retorna o valor 1
3 >>> segundo_elemento = numeros[1]  # Retorna o valor 2
```

Além disso, você pode usar índices negativos para acessar elementos a partir do final da lista. O índice -1 refere-se ao último elemento, o índice -2 refere-se ao penúltimo elemento e assim por diante. Por exemplo:

```
1 >>> numeros = [1, 2, 3, 4, 5]
2 >>> ultimo_elemento = numeros[-1]  # Retorna o valor 5
3 >>> penultimo_elemento = numeros[-2] # Retorna o valor 4
```

As listas também suportam várias operações:

- `len(lista)`: Retorna o número de elementos da lista.
- `lista.append(elemento)` : Adiciona um elemento ao final da lista.
- `lista.insert(indice, elemento)` : Insere um elemento em uma posição específica da lista.
- `lista.remove(elemento)`: Remove a primeira ocorrência de um elemento da lista.
- `lista.pop(indice)`: Remove e retorna o elemento em uma posição específica da lista.
- `lista.sort()`: Ordena a lista em ordem crescente.
- `lista.reverse()`: Inverte a ordem dos elementos na lista.
- `elemento in lista`: Verifica se um elemento está presente na lista, retornando `True` ou `False`.
- `lista.index(elemento)`: Retorna o índice da primeira ocorrência de um elemento na lista.

5.1 Exemplos

Criando uma lista de genes.

```
1 >>> genes = ['BRCA1', 'BRCA2', 'MUTYH', 'MLH1']
2 >>> print(genes)
3 ['BRCA1', 'BRCA2', 'MUTYH', 'MLH1']
```

Acessando informações na lista de genes.

```
1 >>> genes[0] #retorna primeiro elemento
2 'BRCA1'
3 >>> genes[0:3] #retorna os tres primeiros elementos
4 ['BRCA1', 'BRCA2', 'MUTYH']
5 >>> len(genes) #retorna tamanho da lista
6 4
7 >>> genes.index('BRCA2') #retorna index do elemento
8 1
9 >>> genes.count('MUTYH') #repeticoes de um elemento
10 1
```

Adicionando genes no fim da lista.

```
1 >>> genes.append('ISN')
2 >>> print(genes)
3 ['BRCA1', 'BRCA2', 'MUTYH', 'MLH1', 'ISN']
```

Adicionando vários elementos ao fim da lista

```
1 >>> genes.extend(['VDR', 'CYP27B1'])
2 >>> print(genes)
3 ['MUTYH', 'MLH1', 'ISN', 'BRCA2', 'BRCA1', 'VDR', 'CYP27B1']
```

Removendo a primeira ocorrência do elemento da lista.

```
1 >>> genes.remove('ISN')
2 >>> print(genes)
3 ['BRCA1', 'BRCA2', 'MUTYH', 'MLH1']
```

Removendo elementos

```
1 >>> genes.pop() # Remove o último elemento da lista.
2 >>> genes.pop(1) # Remove o elemento indicado pela posição.
```

Adicionando elementos em uma posição específica.

```
1 >>> genes.insert(2, 'ISN')
2 >>> print(genes)
3 ['BRCA1', 'BRCA2', 'ISN', 'MUTYH', 'MLH1']
```

Invertendo a ordem dos elementos na lista

```
1 >>> genes.reverse()
2 >>> print(genes)
3 ['MLH1', 'MUTYH', 'ISN', 'BRCA2', 'BRCA1']
```

Ordena os itens numérica ou alfabeticamente.

```
1 >>> genes.sort(reverse=True)
2 >>> print(genes)
3 ['MUTYH', 'MLH1', 'ISN', 'BRCA2', 'BRCA1']
```