

CAPÍTULO 4

Strings

Uma string pode ser definida como uma sequência de caracteres delimitada por aspas simples (‘ ’) ou aspas duplas (“ ”). As strings são objetos imutáveis e não podem ser modificadas depois de criadas. Entretanto, você pode realizar diversas operações para manipulá-las ou extrair informações.

4.1 Criando *strings*

Para criar uma string é simple. Você deve atribuir um valor entre aspas a uma variável. Por exemplo:

```
1 >>> mensagem = "á0l, programador!!"  
2 >>> print(mensagem)  
3 "á0l, programador!!"
```

4.2 Indexação

Em Python, você pode acessar caracteres individuais de uma string usando indexação. Os índices começam em **0** para o primeiro caractere e vão até o comprimento da string menos um (**-1**). Aqui, usamos colchetes [] para acessar caracteres/posições individuais com base em seus índices. Por exemplo:

```
1 >>> primeiro_caractere = minha_string[0] # Saída '0'
2 >>> segundo_caractere = minha_string[1]  # Saída 'l'
3 >>> ultimo_caractere = minha_string[-1]  # Saída '!'
```

A indexação de strings em Python pode ser aplicada a sequências de DNA, RNA ou proteínas da mesma forma que a indexação de strings comuns. Por exemplo:

Acessar o primeiro nucleotídeo de uma sequência de DNA:

```
1 dna = "ATCGATCGATCG"
2 primeiro_nucleotideo = dna[0]
3 print(primeiro_nucleotideo) # Saída: A
```

Acessar o último aminoácido de uma sequência de proteína:

```
1 proteina = "MIPTEKLSR"
2 ultimo_aminoacido = proteina[-1]
3 print(ultimo_aminoacido) # Saída: R
```

Acessar o segundo códon:

```
1 dna = "ATCGATCGATCG"
2 codon = dna[3:6]
3 print(codon) # Saída: GAT
```

4.3 Métodos

Uma série de métodos úteis pode ser conferida na lista seguinte:

- `capitalize()`: Retorna uma cópia da string com o primeiro caractere em maiúscula e os demais em minúscula.
- `casefold()`: Retorna uma versão em minúsculas da string, adequada para comparação de strings que ignoram diferenças de caso.
- `count(substring, start, end)`: Retorna o número de ocorrências de uma substring na string.
- `endswith(suffix, start, end)`: Verifica se a string termina com uma determinada substring e retorna `True` ou `False`.
- `find(substring, start, end)`: Retorna o índice da primeira ocorrência de uma substring na string, ou `-1` se não for encontrada.

- `index(substring, start, end)`: Retorna o índice da primeira ocorrência de uma substring na string, ou gera uma exceção `ValueError` se não for encontrada.
- `isalnum()`: Verifica se a string contém apenas caracteres alfanuméricos e retorna `True` ou `False`.
- `isalpha()`: Verifica se a string contém apenas caracteres alfabéticos e retorna `True` ou `False`.
- `isdecimal()`: Verifica se a string contém apenas dígitos decimais e retorna `True` ou `False`.
- `isdigit()`: Verifica se a string contém apenas dígitos e retorna `True` ou `False`.
- `isnumeric()`: Verifica se a string contém apenas caracteres numéricos e retorna `True` ou `False`.
- `upper()`: Retorna uma versão em maiúsculas da string.
- `lower()`: Retorna uma versão em minúsculas da string.
- `lstrip(characters)`: Retorna uma cópia da string com os caracteres à esquerda removidos.
- `replace(old, new, count)`: Substitui todas as ocorrências de uma substring por outra na string.
- `rfind(substring, start, end)`: Retorna o índice da última ocorrência de uma substring na string, ou -1 se não for encontrada.
- `rindex(substring, start, end)`: Retorna o índice da última ocorrência de uma substring na string, ou gera uma exceção `ValueError` se não for encontrada.
- `rstrip(characters)`: Retorna uma cópia da string com os caracteres à direita removidos.
- `split(separator, maxsplit)`: Divide a string em substrings com base em um separador e retorna uma lista.
- `startswith(prefix, start, end)`: Verifica se a string começa com um determinado prefixo e retorna `True` ou `False`.

- `strip(characters)`: Retorna uma cópia da string com os caracteres à esquerda e à direita removidos.
- `join(iterable)`: Concatena os elementos de um iterável em uma string, usando a string como separador.
- `title()`: Retorna uma versão da string com a primeira letra de cada palavra em maiúscula.

Essa lista inclui apenas alguns dos métodos disponíveis para strings em Python. Existem outros métodos e funcionalidades que podem ser explorados na documentação oficial do Python.