universidade de aveiro
escola superior de tecnologia e gestão de águeda

Introdução à Programação

Strings

cbt@ua.pt - RSI
pop@ua.pt - CS

Strings

Uma string é uma sequência de caracteres.

É possível aceder aos caracteres um de cada vez utilizando o operador parenteses retos:

```
>>> fruta = 'banana’
>>> letra = fruta[1]
```

A expressão entre parenteses retos é chamada de **índice**. O índice indica qual o caractere na sequência a que queremos aceder.

O índice é um deslocamento do início da string e o deslocamento da primeira letra é **zero**.

```
>>> letter = fruit[0]
>>> letter
'b'
```

Strings

Como índice podemos usar uma expressão que contém variáveis e operadores

```
>>> fruit = 'banana'
>>> i = 1
>>> fruit[i]
'a'
>>> fruit[i+1]
‘n’
```

O valor do índice tem que ser um número inteiro. Caso contrário:

```
>>> letter = fruit[1.5]
TypeError: string indices must be integers
```

Len

Função interna que retorna o número de caracteres de uma string

```
>>> fruit = 'banana'
>>> len(fruit)
6
```

Como obter a última letra de uma string?

```
>>> last = fruit[length-1]
>>> last
'a’
```

Como começamos a contar do zero, as seis letras são numeradas de 0 a 5. Para obter o último caractere, você deve subtrair 1 do comprimento

Iteração em Strings - *while*

Processamento de uma string um caractere de cada vez.

```
fruit = 'banana'
index = 0
while index < len(fruit):
    letter = fruit[index]
    print(letter)
    index = index + 1
```

A condição do loop é index < len(fruit). Quando index é igual ao comprimento da string, a condição é falsa e o corpo do loop não é executado.

Iteração em Strings - *for*


```
fruit = 'banana'
for letter in fruit:
    print(letter)
```

A cada iteração do ciclo é atribuído à variável letter o caracter seguinte.

O ciclo termina quando não existirem mais carateres na string para serem iterados.

Sub-strings - Slice

Um segmento de uma string é chamado de slice. A seleção é semelhante à seleção de um caractere

```
>>> s = 'Monty Python'
>>> s[0:5]
'Monty'
>>> s[6:12]
' Python'
```

O operador [n:m] retorna a parte da string do caractere "n" até ao caractere "m", incluindo o primeiro, mas excluindo o último.

Podemos omitir o primeiro índice (antes dos dois pontos), a sub-string começará no início da string.

```
>>> fruit = 'banana'
>>> fruit[:3]
'ban'
```

Strings são imutáveis

O que significa que não se pode alterar uma string existente.

É tentador usar o operador [] com a intenção de alterar um caractere de uma string

```
>>> greeting = 'Hello, world!'
>>> greeting[0] = 'J'
TypeError: 'str' object does not support item assignment
```

Métodos das Strings

universidade de aveiro
escola superior de tecnologia e gestão de águeda

As Strings fornecem métodos que executam uma variedade de operações úteis.

Um método recebe argumentos e retorna um valor.

upper

Pega numa string e retorna uma nova string com todas as letras maiúsculas.

```
>>> word = 'banana'
>>> new_word = word.upper()
>>> new_word
'BANANA'
```

Métodos das Strings

find

Devolve o índice da primeira aparição do caractere na string.

```
>>> word = 'banana'
>>> index = word.find('a')
>>> index
1
```

O método find funciona não só com caracteres mas também com substrings.

```
>>> word = 'banana'
>>> index = word.find('a')
>>> index
1
```

Operador in

in é um operador booleano, o que significa que devolve True ou False.

```
>>> 'a' in 'banana'
True
>>> 'seed' in 'banana'
False
```

O seguinte exempo imprime todas as letras da string word1 que existem na string word2

```
for letter in word1:
    if letter in word2:
        print(letter)
```

Comparação de Strings

Os operadores relacionais também funcionam em strings.

Para ver se duas strings são iguais:

```
if word == 'banana’:
    print('All right, bananas.’)
```

Outras operações relacionais são úteis para colocar palavras por ordem alfabética:

```
if word < 'banana’:
    print('Your word, ' + word + ', comes before banana.')
elif word > 'banana’:
    print('Your word, ' + word + ', comes after banana.')
else:
    print('All right, bananas.')
```

Comparação de Strings

universidade de aveiro
escola superior de tecnologia e gestão de águeda

O Python não lida com letras maiúsculas e minúsculas da mesma forma que as pessoas.

Todas as letras maiúsculas vêm antes de todas as letras minúsculas, então:

Your word, Pineapple, comes before banana.

Uma maneira comum de resolver esse problema é converter strings para um formato padrão, como por exemplo todas as letras em minúsculas, antes de realizar a comparação.