

# Chapter 6

## Strings

### A string is a sequence

- Chuỗi gồm nhiều ký tự, mỗi ký tự có chỉ số (index) bắt đầu từ 0.
- `fruit[0]` → ký tự đầu tiên ('b').
- `fruit[-1]` → ký tự cuối ('a').

01

```
1 fruit = 'banana'
2 print(fruit[0]) # b
```

### Getting the length of a string using len

02 : Trả về số ký tự trong chuỗi.

Thường dùng khi duyệt qua chuỗi bằng chỉ số.

Thường kết hợp với vòng lặp while:

```
1 index = 0
2 while index < len(fruit):
3     print(fruit[index])
4     index += 1
```

### Traversal through a string with a loop

Dùng for ngắn gọn hơn, ít lỗi hơn.

```
1 for ch in 'banana':
2     print(ch)
```

### String slices

Chỉ lấy đến `end - 1`, không bao gồm ký tự `end`.

04

```
1 s = 'Monty Python'
2 print(s[0:5]) # Monty
3 print(s[6:]) # Python
```

Nếu bỏ `start` → mặc định là 0, bỏ `end` → đến cuối chuỗi:

05

### Strings are immutable

Không thể thay đổi ký tự trực tiếp

```
1 greet = 'Hello'
2 # greet[0] = 'J' X
3 greet = 'J' + greet[1:] # 'Jello'
```

### Phân tích chuỗi (Parsing)

Dùng `find()` và slicing để trích thông tin:

```
1 data = 'From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008'
2 atpos = data.find('@')
3 spos = data.find(' ', atpos)
4 host = data[atpos+1 : spos]
5 print(host) # uct.ac.za
```

10

### string methods

`['capitalize', 'casefold', 'center', 'count', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'format_map', 'index', 'isalnum', 'isalpha', 'decimal', 'isdigit', 'isidentifier', 'islower', 'isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'maketrans', 'partition', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']`

09

### String comparison

So sánh theo thứ tự bảng chữ cái (lexicographical order).

Chữ hoa < chữ thường theo mã ASCII.

```
1 word = 'Pineapple'
2 if word < 'banana':
3     print("comes before banana")
```

08

### The in operator

```
1 'a' in 'banana' # True
2 'seed' in 'banana' # False
```

Dùng để kiểm tra chuỗi con có trong chuỗi chính không.

07

### Looping and counting

```
1 word = 'banana'
2 count = 0
3 for letter in word:
4     if letter == 'a':
5         count += 1
6 print(count)
```

06

Có thể dùng `word.count('a')` cho ngắn.