

# CHEATSHEET chương 6

## 1 – A string is a sequence

String là một chuỗi ký tự có thể duyệt từng ký tự như một sequence

vd:

```
1 fruit = 'banana'  
2 print(fruit[1])  
3  
4
```

kết quả

```
DELL/New folder/huhu.py/999/61.py > ...  
a
```

## 3- Duyệt chuỗi bằng vòng lặp

Có thể duyệt chuỗi bằng:

- **for:** đơn giản, phổ biến
- **while:** duyệt theo chỉ số

dùng lệnh:while

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

lệnh :for

```
1.py > ...  
2. for char in 'banana':  
3.     print(char)
```

kết quả:

```
DELL/New folder/huhu.py/999/61.py > ...  
b  
a  
n  
a  
n  
a
```

## 2 – LẤY ĐỘ DÀI CHUỖI BẰNG LEN()

TRÄ VỀ SỐ KÝ TỰ TRONG CHUỖI.

NẾU TRUY CẬP NGOÀI PHẠM VI ĐỘ DÀI → LỖI.

```
1.py > ...  
fruit = 'banana'  
length = len(fruit)  
print("Độ dài của chuỗi:", length)
```

```
DELL/New folder/huhu.py/999/61.py > ...  
Độ dài của chuỗi: 6
```

## 4- Slicing – s[start:end]

- Dùng để cắt ra một phần của chuỗi

```
1. 61.py > ...  
2. s = "Monty Python"  
3. s[0:5]  
4. s[6:]  
5. print(s[:5])  
6. print(s[6:])
```

```
PS C:\Users\DELL\New folder\DELL>New folder/huhu.py/999/61.py > ...  
Monty  
Python
```

## 5 – Chuỗi là bất biến

- Immutable = không thay đổi được.
- Bạn không thể sửa 1 ký tự trong chuỗi như sửa phần tử của list.
- Cách tốt nhất bạn có thể làm là tạo một chuỗi mới là

vd:

```
1 greeting = "Hello, world!"  
2 new_greeting = "J" + greeting[1:]  
3 print(new_greeting)
```

```
1,337,3,DELL,New folder\huhu.py,999,81,py  
Jello, world!  
PS C:\Users\DELL\New folder\huhu.py\999>
```

## 6-( in)Lặp và đếm ký tự

Dùng để kiểm tra chuỗi con có xuất hiện trong

chuỗi lớn hay không.

thường dùng:

- Tìm từ khóa
- Kiểm tra sự tồn tại
- Điều kiện trong if

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

```
1,337,3,DELL,New folder\huhu.py,999,81,py  
3
```

## 7-So sánh chuỗi <, >, ==

Dùng để so sánh chuỗi theo thứ tự từ điển.

Lý do dùng:

- Sắp xếp chuỗi
- Kiểm tra điều kiện
- Viết thuật toán so sánh dữ liệu text

```
1 print("apple" < "banana")  
:/Users/DELL/New folder\huhu.py\999  
True
```

## 8-upper() – Dùng để chuyển chuỗi thành chữ in hoa.

Dùng khi:

- Muốn chuẩn hóa dữ liệu
- So sánh chuỗi không phân biệt chữ hoa/thường
- lower() – Dùng để chuyển chuỗi thành chữ thường.

```
1 username = "NgUyeN vAn a"  
2 # Chuyển thành chữ IN HOA  
3 upper_name = username.upper()  
4 # Chuyển thành chữ thường  
5 lower_name = username.lower()  
6 print("Tên in hoa:", upper_name)  
7 print("Tên in thường:", lower_name)  
8
```

Tên in hoa: NGUYEN VAN A

Tên in thường: nguyen van a

## 9-strip() – Dùng để xóa khoảng trắng ở đầu và cuối chuỗi.

Hay dùng khi:

- Xử lý input người dùng
- Tách dòng dữ liệu sạch hơn (data cleaning)

```
user_input = "    Nguyen Van A    "  
clean_name = user_input.strip()  
  
print("Tên sau khi xử lý:", clean_name)
```

```
1,337,3,DELL,New folder\huhu.py,999,81,py  
Tên sau khi xử lý: Nguyen Van A
```

## 10-find(x) – Dùng để tìm vị trí xuất hiện đầu tiên của x trong chuỗi.

Nếu không tìm thấy → trả về -1.

Dùng khi:

- Tìm vị trí ký tự
- Tách dữ liệu bằng slicing
- Kiểm tra dữ liệu thô (email, log, text)

```
ex6.py > ...  
1 email = "user123@example.com"  
2 at_pos = email.find("@")  
3 username = email[:at_pos]  
4 print(username) # Kết quả: user123
```

```
1,337,3,DELL,New folder\huhu.py,999,81,py  
user123
```

### 11-replace(a, b) –

- Dùng để thay tất cả chuỗi a bằng b.

```
text = "I love Python. Python is fun!"
result = text.replace("Python", "Java")
print(result)
```

I love Java. Java is fun!

### 12- split() –

Dùng để tách chuỗi thành list theo khoảng trắng (hoặc ký tự khác).

```
exb.py > ...
1  data = "apple,banana,orange"
2  items = data.split(",")
3  print(items)
```

['apple', 'banana', 'orange']

### 13. Debugging (gỡ lỗi)

- Dùng để tránh lỗi truy cập chuỗi rỗng.

```
line = ""

# Kiểm tra trước khi truy cập index 0
if len(line) > 0 and line[0] == '#':
    print("Là dòng comment")
else:
    print("Không phải comment hoặc chuỗi rỗng")
```

Không phải comment hoặc chuỗi rỗng  
PS C:\Users\DELL\New folder\huhu.py\999>

### 14.-Parsing (tách dữ liệu)

- Dùng để trích thông tin từ chuỗi bằng find + slice.

```
1  data = "From stephen.marquard@uct.ac.za Sat Jan 5 "
2  # tìm vị trí dấu @
3  at_pos = data.find("@")
4  # tìm khoảng trắng đầu tiên SAU dấu @
5  space_pos = data.find(" ", at_pos)
6  # lấy domain bằng slicing
7  domain = data[at_pos + 1 : space_pos]
8  print("Domain:", domain)
```

Domain: uct.ac.za

PS C:\Users\DELL\New folder\huhu.py\999>

### 15-Format operator %

- Dùng để nhúng (chèn) dữ liệu vào chuỗi

```
1  name = "Alice"
2  score = 95
3
4  message = "Xin chào %s, điểm của bạn là %d" % (name, score)
5
6  print(message)
7
```

hu.py/999/ex6.py"

Xin chào Alice, điểm của bạn là 95

**Exercise 5:** Take the following Python code that stores a string: ‘str = ’X-DSPAM-Confidence:0.8475’ Use find and string slicing to extract the portion of the string after the colon character and then use the float function to convert the extracted string into a floating point number

```
1  str = 'X-DSPAM-Confidence:0.8475'
2  # Tìm vị trí dấu hai chấm
3  colon_pos = str.find(':')
4  # Lấy phần sau dấu hai chấm (bỏ ký tự :)
5  number_str = str[colon_pos + 1 :]
6  # Chuyển sang float
7  number = float(number_str)
8  print(number)
9
10
```

```
PS C:\Users\DELL\New folder\huhu.py\999> & C:/Users/DEL
L/AppData/Local/Programs/Python/Python313/python.exe "c
:/Users/DELL/New folder/huhu.py/999/ex6.py"
0.8475
PS C:\Users\DELL\New folder\huhu.py\999>
```