

Kiểu dữ liệu XÂU KÍ TỰ – XỬ LÝ XÂU KÍ TỰ

Học xong bài này, em sẽ:

- ✓ Nhận biết được dữ liệu kiểu xâu.
- ✓ Viết được câu lệnh Python trích xâu con từ xâu cho trước.
- ✓ Sử dụng được một số phép xử lý xâu thường dùng trong Python.



Em đã từng sử dụng phần mềm xử lý văn bản. Theo em, trong ngôn ngữ lập trình, ngoài kiểu dữ liệu số có cần một kiểu dữ liệu không phải là số dùng cho các bài toán xử lý văn bản hay không? Nếu có kiểu dữ liệu như vậy thì nên có những phép xử lý nào trên dữ liệu thuộc kiểu đó?

1 Kiểu dữ liệu xâu kí tự



1

Em hãy đọc chương trình sau đây và cho biết mỗi biến: `so_hop`, `khoi_luong_hop`, `don_vi_kl` chứa dữ liệu thuộc kiểu nào?

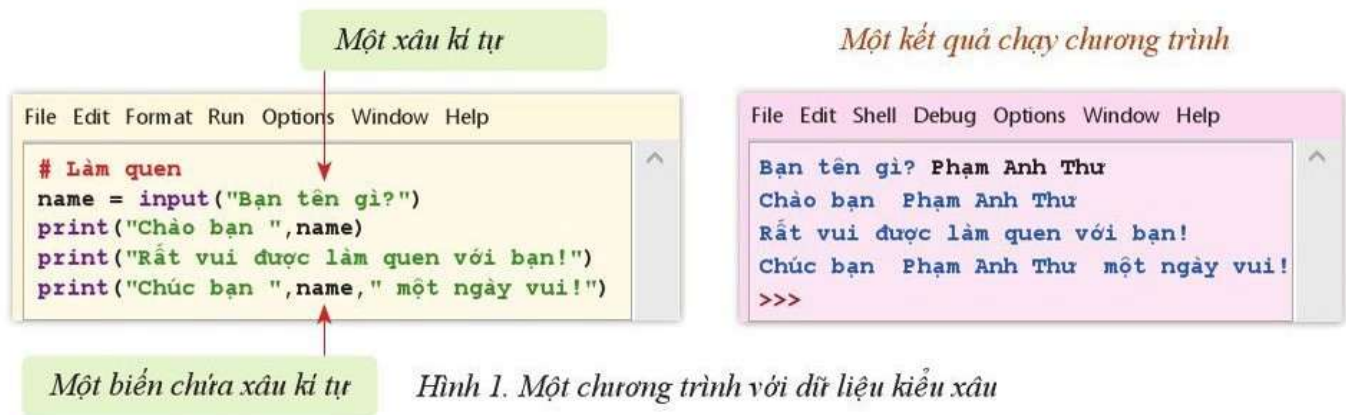
File Edit Format Run Options Window Help

```
# Tính khối lượng cafe trong bao
so_hop = int(input(" Số hộp cafe trong bao: "))
khoi_luong_hop = float(input(" Mỗi hộp nặng: "))
don_vi_kl = input(" Đơn vị tính khối lượng: ")
print(" Khối lượng cafe trong bao là:", so_hop*khoi_luong_hop, don_vi_kl)
```

Gợi ý: có thể dùng hàm `type()` để kiểm tra kết quả.

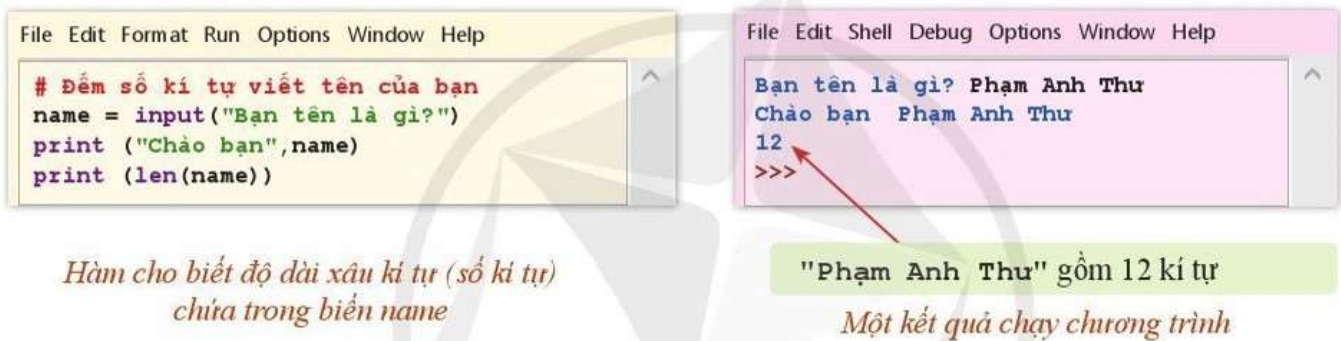
Để giải quyết các bài toán trong thực tế gồm cả dữ liệu số và không phải là số, các ngôn ngữ lập trình bậc cao đều cho chúng ta dùng các biến thuộc kiểu dữ liệu xâu kí tự và cung cấp một số công cụ để xử lý dữ liệu kiểu xâu kí tự. Một xâu kí tự là một dãy các kí tự. Trong Python, xâu kí tự được đặt trong cặp nháy đơn (hoặc nháy kép).

Ví dụ 1. Hình 1 minh họa một chương trình sử dụng kiểu dữ liệu xâu kí tự và một biến có chứa xâu kí tự.



Hình 1. Một chương trình với dữ liệu kiểu chuỗi

Các ký tự trong chuỗi được đánh số bắt đầu từ 0. Python cung cấp hàm `len()` để đếm số ký tự trong một chuỗi kể cả ký tự dấu cách. Số ký tự trong chuỗi được gọi là độ dài của chuỗi. Hình 2 minh họa một chương trình sử dụng hàm `len()` và kiểu dữ liệu chuỗi ký tự.



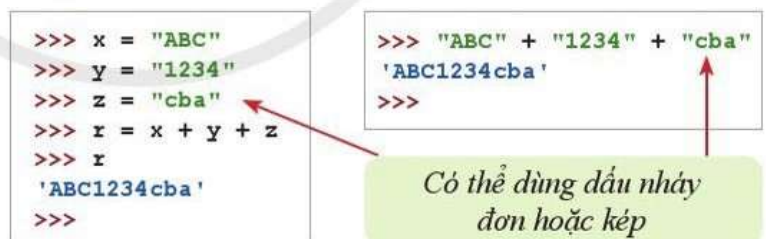
Hình 2. Một chương trình sử dụng hàm `len()`

2 Một số hàm xử lý chuỗi ký tự

Python cung cấp nhiều công cụ để xử lý chuỗi. Một số công cụ thường dùng là:

a) Ghép chuỗi bằng phép +

Viết liên tiếp các chuỗi cần ghép theo thứ tự và đặt giữa hai chuỗi kề nhau dấu “+” (Hình 3).



Hình 3. Một ví dụ về ghép chuỗi

b) Đếm số lần xuất hiện chuỗi con

Hàm `y.count(x)` đếm số lần xuất hiện không giao nhau của `x` trong `y` (Hình 4).



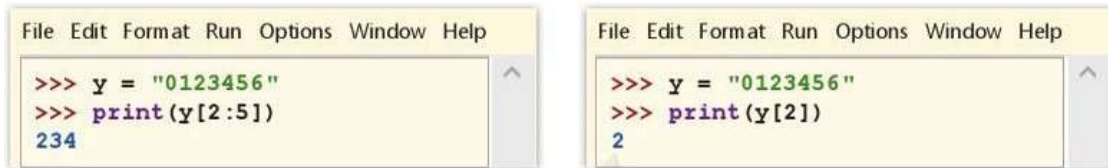
Hình 4. Số lần xuất hiện chuỗi con

Có thể nêu các tham số xác định cụ thể phạm vi tìm kiếm. Ví dụ:

- `y.count(x, 3)` cho biết số lần xuất hiện các xâu `x` không giao nhau trong xâu `y` nhưng chỉ trong phạm vi từ kí tự thứ ba đến kí tự cuối của xâu `y`.
- `y.count(x, 3, 5)` cho biết số lần xuất hiện các xâu `x` không giao nhau trong xâu `y` nhưng chỉ trong phạm vi từ kí tự thứ ba đến kí tự thứ năm của xâu `y`.

c) Xác định xâu con

Xác định xâu con của xâu `y` từ vị trí `m` đến trước vị trí `n` ($m < n$) ta có cú pháp: `y[m:n]` (Hình 5).



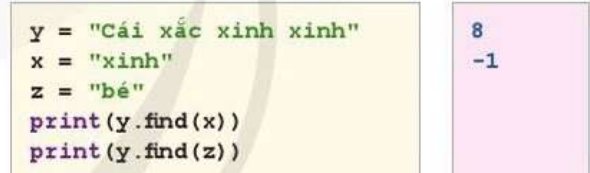
Hình 5. Xác định một xâu con

Các trường hợp đặc biệt:

- `y[:m]` là xâu con gồm `m` kí tự đầu tiên của xâu `y`.
- `y[m:]` là xâu con nhận được bằng cách bỏ `m` kí tự đầu tiên của xâu `y`.

d) Tìm vị trí xuất hiện lần đầu tiên của một xâu trong xâu khác

Hàm `y.find(x)` trả về số nguyên xác định vị trí đầu tiên trong xâu `y` mà từ đó xâu `x` xuất hiện như một xâu con của xâu `y`. Nếu xâu `x` không xuất hiện như một xâu con, kết quả trả về sẽ là `-1`.



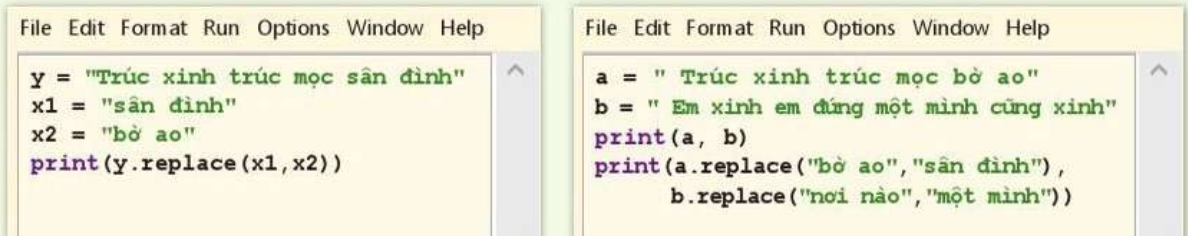
Hình 6. Tìm vị trí đầu tiên của một xâu con

e) Thay thế xâu con

Hàm `y.replace(x1, x2)` tạo xâu mới từ xâu `y` bằng cách thay thế xâu con `x1` của `y` bằng xâu `x2`. Tất cả các xâu con bằng `x1` và không giao nhau của `y` đều được thay bằng xâu `x2`.



Em hãy đọc các chương trình sau đây và cho biết kết quả nhận được khi thực hiện chương trình.





Bài 1. Hãy dự đoán kết quả đưa ra màn hình sau mỗi câu lệnh xuất dữ liệu **print()** trong chương trình ở hình bên và sau đó dùng cửa sổ Shell để đối chiếu, kiểm tra từng kết quả dự đoán.

```
File Edit Format Run Options Window Help
xau1 = "Hà Nội là thủ đô của nước Việt Nam."
xau2 = "Nam Khánh sinh ra ở Hà Nội."
xau = xau1 + xau2
print (xau) -----> ?
print (xau.count ("N", 6)) -----> ?
print (xau.find ("Khánh")) -----> ?
print (xau[25:34]) -----> ?
print (xau.replace ("Khánh", "An")) -----> ?
```

Bài 2. Em hãy viết chương trình nhập từ bàn phím chuỗi ghi ngày tháng dạng dd/mm/yyyy, trong đó dd là hai ký tự chỉ ngày, mm là hai ký tự chỉ tháng, yyyy là bốn ký tự chỉ năm. Sau đó đưa ra màn hình ngày, tháng, năm dưới dạng chuỗi "Ngày dd tháng mm năm yyyy".

Ví dụ:

INPUT	OUTPUT
15/12/2022	Ngày 15 tháng 12 năm 2022



Nhập vào từ bàn phím hai chuỗi s1 và s2, mỗi chuỗi không chứa ký tự dấu cách ở đầu và cuối chuỗi cũng như không chứa hai hay nhiều dấu cách liên tiếp nhau. Nếu chuỗi không chứa dấu cách thì nó là một từ, trong trường hợp ngược lại, dấu cách là dấu phân tách các từ trong chuỗi. Ví dụ, chuỗi "Bước tới Đèo Ngang, bóng xế tà" chứa bảy từ. Em hãy viết chương trình xác định và đưa ra màn hình tổng số từ trong hai chuỗi s1 và s2 đã cho.

Ví dụ:

INPUT	OUTPUT
Dưới trăng quỳên đã gọi hè Đầu tường lửa lựu lập loè đâm bông	14



Trong các câu sau đây, những câu nào đúng?

- 1) Có thể ghép các chuỗi để được chuỗi mới.
- 2) Có thể tìm vị trí một chuỗi con trong một chuỗi.
- 3) Không thể xóa một chuỗi con trong một chuỗi.
- 4) Không thể đếm số lần xuất hiện một chuỗi con trong một chuỗi.

Tóm tắt bài học

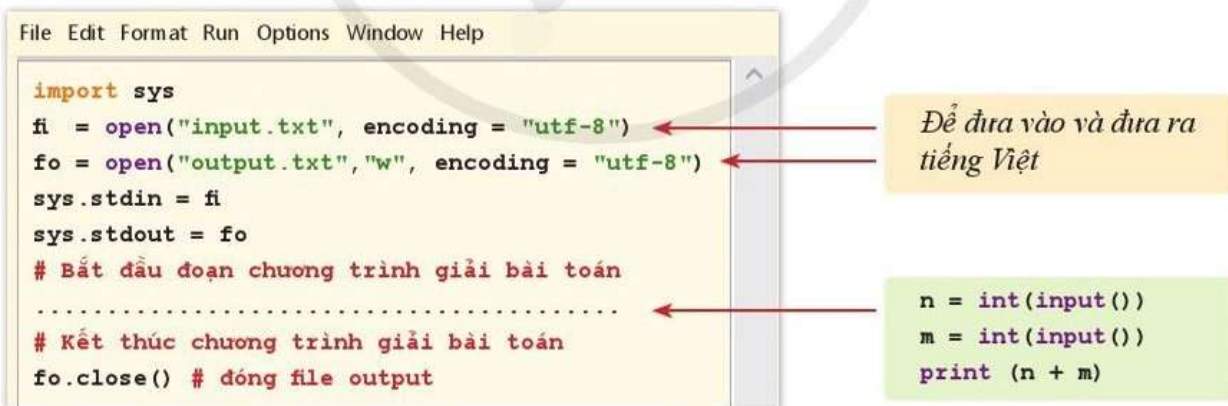
- ✓ Trong các ngôn ngữ lập trình bậc cao có kiểu dữ liệu chuỗi ký tự và các chương trình cung cấp thao tác xử lý chuỗi ký tự.
- ✓ Trong Python, phép “+” dùng để ghép nối các chuỗi.
- ✓ Trong Python, có một số hàm xử lý chuỗi thường dùng: xác định độ dài chuỗi, đếm số lần xuất hiện chuỗi con, tìm vị trí xuất hiện lần đầu tiên của một chuỗi trong chuỗi khác, thay thế chuỗi con và cách xác định chuỗi con.

BÀI TÌM HIỂU THÊM

THAY THẾ THIẾT BỊ VÀO – RA CHUẨN

Giống như nhiều ngôn ngữ lập trình khác, Python mặc định sử dụng bàn phím làm thiết bị cho nhập dữ liệu vào (**stdin**) và màn hình làm thiết bị xuất dữ liệu ra (**stdout**). Như vậy, bàn phím và màn hình là thiết bị vào – ra chuẩn.

Khi dữ liệu vào – ra lớn, các thiết bị này không còn phù hợp trong việc thực hiện chương trình cũng như gỡ lỗi. Python cho phép thay thiết bị chuẩn bằng file văn bản. Ví dụ, dữ liệu nhập vào được chuẩn bị trong file `input.txt` (bằng `notepad` hay bằng chính chương trình soạn thảo của Python), kết quả sẽ được đưa ra file văn bản `output.txt`, việc thay thế thiết bị chuẩn được thực hiện theo mẫu sau:



The image shows a Python script in a text editor with a menu bar (File, Edit, Format, Run, Options, Window, Help). The script is as follows:

```
import sys
fi = open("input.txt", encoding = "utf-8")
fo = open("output.txt", "w", encoding = "utf-8")
sys.stdin = fi
sys.stdout = fo
# Bắt đầu đoạn chương trình giải bài toán
.....
# Kết thúc chương trình giải bài toán
fo.close() # đóng file output
```

Annotations with arrows point to specific lines:

- Two arrows point to the `open` calls for `input.txt` and `output.txt` with the text: "Để đưa vào và đưa ra tiếng Việt".
- An arrow points to the `.....` line with the text: `n = int(input())`
`m = int(input())`
`print (n + m)`

Chẳng hạn, nếu đưa ba dòng lệnh nhập `n` và `m` vào và sau đó in ra tổng của chúng thì với file `input.txt` (Hình 1a), ta sẽ nhận được file `output.txt` (Hình 1b).

input
5
3

Hình 1a

output
8

Hình 1b

Lưu ý: Tên file trong các câu lệnh `open` và các tên biến `fi`, `fo` là tùy chọn.