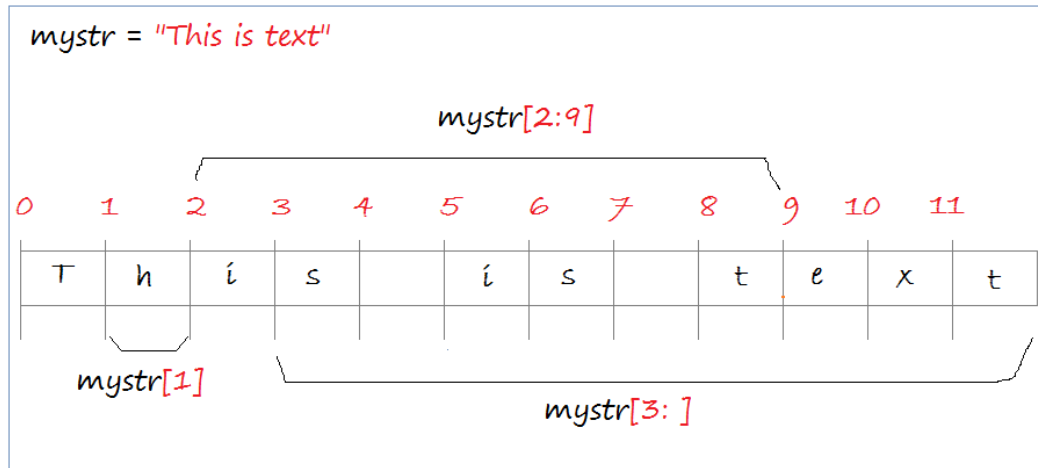




## LAB 8 : CHUỖI – ‘STRING’

### A. HIỂU VỀ CHUỖI

Python không hỗ trợ kiểu ký tự (Character type), ký tự được coi là một string với độ dài 1. Các ký tự trong string được đánh chỉ số bắt đầu từ 0. Bạn có thể truy cập vào các chuỗi con (substring) thông qua chỉ số.



### B. CÁC HÀM TRONG CHUỖI

1. **Hàm `strip()`** : loại bỏ bất kỳ khoảng trắng từ đầu hoặc cuối cùng.

Ví dụ:

```
a = " Hello World! "  
print(a.strip()) # returns "Hello World!"
```

2. **Hàm `lower()`** : trả về chuỗi chữ thường

Ví dụ:

```
a = "Hello World!"  
print(a.lower()) # returns "hello world!"
```

3. **Hàm `replace()`**: thay thế một chuỗi bằng một chuỗi khác

Ví dụ:

```
a = "Hello World!"  
print(a.replace("l", "t")) # returns "hetto wordt!"
```

4. **Hàm `split()`**: tách chuỗi thành các chuỗi con



Ví dụ:

```
a = "Hello, World!"  
print(a.split(",")) # returns "'Hello', ' World!'"
```

## 5. Hàm **count()** : Hàm này có tác dụng đếm xem trong chuỗi có bao nhiêu ký tự cần tìm

```
string.count(sub, start, end)
```

- **sub** là chuỗi các bạn cần tìm kiếm và đếm.
- **start** là index bắt đầu của chuỗi cần tìm. Mặc định thì **start = 0**.
- **end** là index kết thúc của chuỗi cần tìm. Mặc định thì **end = len()** của chuỗi.

Ví dụ:

```
string = "toidicode.com"  
print(string.count('i'));  
# Kết quả: 2  
print(string.count('i', 3));  
# Kết quả: 1
```

## 6. Kiểm tra sự tồn tại của chuỗi trong chuỗi

Để kiểm tra xem một cụm từ hoặc ký tự nào đó có trong một chuỗi hay không, chúng ta có thể sử dụng các từ khóa **in** hoặc **not in**.

Ví dụ:

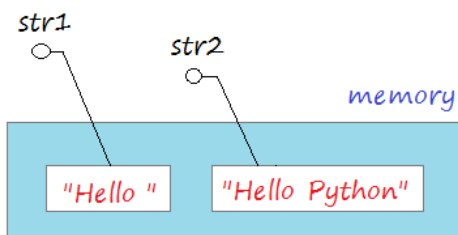
```
str1 = "hoc lap trinh python string."  
x = "python" in str1  
print(x)  
True # có tồn tại ký tự 'python' trong chuỗi str1  
  
str1 = "hoc lap trinh python string."  
x = "pythopp" in str1  
print(x)  
False # không tồn tại ký tự 'pythopp' trong chuỗi str1
```

## 7. Nối chuỗi trong Python :

Để nối hoặc kết hợp hai chuỗi với nhau bạn có thể sử dụng toán tử +.



```
str1 = "Hello "  
str2 = str1 + "Python"
```



Ví dụ:

```
Str1 = "Hello "  
Str2 = str1 + "Python "  
print(str2)  
# returns "Hello Python"
```

## 8. Định dạng chuỗi trong Python

Trong Python, chúng ta không thể kết hợp các chuỗi và số như thế này:

```
n = 3  
txt = "Hello Python " + 3  
print(txt)
```

Chúng ta có thể kết hợp chuỗi và số bằng cách sử dụng hàm **format()**.

```
n = 3  
txt = "Hello Python {}"  
print(txt.format(n))
```

# kết quả => Hello Python 3

Ví dụ thêm

```
hoten = 'Tony Teo'  
namsinh = '2000'  
mssv = '1234567'  
khoa = 'CNTT'  
  
thongtin = 'Họ Tên:{}, Năm Sinh:{}, Mã Số Sinh Viên:{}, Khoa:{}' # sắp xếp theo thứ tự các chuỗi  
print(thongtin.format(hoten,namsinh,mssv,khoa))
```

## 9. Các phương thức và hàm đã xây dựng sẵn để xử lý chuỗi trong Python



1	<u>Hàm capitalize()</u> Viết hoa chữ cái đầu tiên của chuỗi
2	<u>Hàm center(width, fillchar)</u> Trả về một chuỗi mới, trong đó chuỗi ban đầu đã được cho vào trung tâm và hai bên đó là các fillchar sao cho tổng số ký tự của chuỗi mới là width
3	<u>Hàm count(str, beg= 0,end=len(string))</u> Đếm xem chuỗi str này xuất hiện bao nhiêu lần trong chuỗi string hoặc chuỗi con của string nếu bạn cung cấp chỉ mục ban đầu start và chỉ mục kết thúc end
4	<u>Hàm endswith(suffix, beg=0, end=len(string))</u> Xác định xem nếu chuỗi string hoặc chuỗi con đã cho của string (nếu bạn cung cấp chỉ mục bắt đầu beg và chỉ mục kết thúc end) kết thúc với hậu tố suffix thì trả về true, nếu không thì phương thức này trả về false
5	<u>Hàm expandtabs(tabsize=8)</u> Mở rộng các tab trong chuỗi tới số khoảng trống đã cho; mặc định là 8 space cho mỗi tab nếu bạn không cung cấp tabsize
6	<u>Hàm find(str, beg=0 end=len(string))</u> Xác định xem chuỗi str có xuất hiện trong chuỗi string hoặc chuỗi con đã cho của string (nếu bạn cung cấp chỉ mục bắt đầu beg và chỉ mục kết thúc end), nếu xuất hiện thì trả về chỉ mục của str, còn không thì trả về -1
7	<u>Hàm index(str, beg=0, end=len(string))</u> Tương tự như find(), nhưng tạo ra một ngoại lệ nếu str là không được tìm thấy
8	<u>Hàm isalnum()</u> Trả về true nếu chuỗi có ít nhất một ký tự và tất cả ký tự là chữ-số. Nếu không hàm sẽ trả về false
9	<u>Hàm isalpha()</u> Trả về true nếu chuỗi có ít nhất 1 ký tự và tất cả ký tự là chữ cái. Nếu không phương thức sẽ trả về false
10	<u>Hàm isdigit()</u> Trả về true nếu chuỗi chỉ chứa các chữ số, nếu không là false
11	<u>Hàm islower()</u> Trả về true nếu tất cả ký tự trong chuỗi là ở dạng chữ thường, nếu không là false



12	<u>Hàm <code>isnumeric()</code></u> Trả về true nếu một chuỗi dạng Unicode chỉ chứa các ký tự số, nếu không là false
13	<u>Hàm <code>isspace()</code></u> Trả về true nếu chuỗi chỉ chứa các ký tự khoảng trắng whitespace, nếu không là false
14	<u>Hàm <code>istitle()</code></u> Trả về true nếu chuỗi là ở dạng titlecase, nếu không là false
15	<u>Hàm <code>isupper()</code></u> Trả về true nếu tất cả ký tự trong chuỗi là chữ hoa
16	<u>Hàm <code>join(seq)</code></u> Nối chuỗi các biểu diễn chuỗi của các phần tử trong dãy seq thành một chuỗi
17	<u>Hàm <code>len(string)</code></u> Trả về độ dài của chuỗi
18	<u>Hàm <code>ljust(width[, fillchar])</code></u> Trả về một chuỗi mới, trong đó có chuỗi ban đầu được căn chỉnh vào bên trái và bên phải là các fillchar sao cho tổng số ký tự là width
19	<u>Hàm <code>lower()</code></u> Chuyển đổi tất cả chữ hoa trong chuỗi sang kiểu chữ thường
20	<u>Hàm <code>lstrip()</code></u> Xóa tất cả các khoảng trống trắng ban đầu (leading) trong chuỗi
21	<u>Hàm <code>max(str)</code></u> Trả về ký tự chữ cái lớn nhất từ chuỗi str đã cho
22	<u>Hàm <code>min(str)</code></u> Trả về ký tự chữ cái nhỏ nhất từ chuỗi str đã cho
23	<u>Hàm <code>replace(old, new [, max])</code></u> Thay thế tất cả sự xuất hiện của old trong chuỗi với new với số lần xuất hiện max (nếu cung cấp)
24	<u>Hàm <code>rfind(str, beg=0, end=len(string))</code></u> Tương tự hàm <code>find()</code> , nhưng trả về chỉ mục cuối cùng
25	<u>Hàm <code>rindex( str, beg=0, end=len(string))</code></u> Giống <code>index()</code> , nhưng trả về chỉ mục cuối cùng nếu tìm thấy



26	<u><a href="#">Hàm rjust(width,[, fillchar])</a></u> Trả về một chuỗi mới, trong đó có chuỗi ban đầu được căn chỉnh vào bên phải và bên trái là các fillchar sao cho tổng số ký tự là width
27	<u><a href="#">Hàm.rstrip()</a></u> Xóa bỏ tất cả các khoảng trống trắng ở cuối (trailing) của chuỗi
28	<u><a href="#">Hàm.split(str="", num=string.count(str))</a></u> Chia chuỗi theo delimiter đã cho (là space nếu không được cung cấp) và trả về danh sách các chuỗi con; nếu bạn cung cấp num thì chia chuỗi thành num chuỗi con
29	<u><a href="#">Hàm.splitlines( num=string.count('\n'))</a></u> Trả về một List gồm tất cả các dòng trong chuỗi, và tùy ý xác định các ngắt dòng (nếu num được cung cấp và là true).
30	<u><a href="#">Hàm.startswith(str, beg=0,end=len(string))</a></u> Xác định xem chuỗi hoặc chuỗi con (nếu bạn cung cấp chỉ mục bắt đầu beg và chỉ mục kết thúc end) có bắt đầu với chuỗi con str không, nếu có trả về true, nếu không là false
31	<u><a href="#">Hàm.strip([chars])</a></u> Thực hiện cả hai phương thức lstrip() và rstrip() trên chuỗi
32	<u><a href="#">Hàm.swapcase()</a></u> Đảo ngược kiểu của tất cả ký tự trong chuỗi
33	<u><a href="#">Hàm.title()</a></u> Trả về một bản sao của chuỗi trong đó tất cả ký tự đầu tiên của tất cả các từ là ở kiểu chữ hoa.
34	<u><a href="#">Hàm.upper()</a></u> Chuyển đổi các chữ thường trong chuỗi thành chữ hoa
35	<u><a href="#">Hàm.zfill (width)</a></u> Trả về một chuỗi mới, trong đó bao gồm chuỗi ban đầu và được đệm thêm với các số 0 vào bên trái sao cho tổng ký tự là width
36	<u><a href="#">Hàm.isdecimal()</a></u> Trả về true nếu một chuỗi dạng Unicode chỉ chứa các ký tự thập phân, nếu không là false

### C. VÍ DỤ



```
#=====
#                               CÁC VẤN ĐỀ LIÊN QUAN ĐẾN CHUỖI
#=====
chuoai = 'hello python' # chuỗi có 12 ký tự
chuoai_1 = ' How are you'
#=====
#                               KIỂM TRA VÀ IN CÁC KÝ TỰ THEO Ý MUỐN
#=====

print('he' in chuoai) # kiểm tra ký tự 'he' có trong chuỗi hiện True
print('sao' in chuoai) # kiểm tra ký tự 'sao' không có trong chuoai, hiện ra False
print(chuoai[-1]) # in ký tự cuối cùng là 'n'
print(chuoai[1:5]) # in ký tự thứ 2 (thứ 1 là 'h') đến thứ 5 là 'ello'
print(chuoai[5:-2]) # in ký tự thứ 5 và bỏ đi 2 ký tự cuối là 'pyth'
print('-'*70)
#=====
#                               HÀM CHO CHUỖI
#=====

a = len(chuoai) # Tính chiều dài chuỗi là 12
print('chiều dài chuỗi là:',a)
b = str.upper(chuoai) # Thành chữ in HOA
print('hello python viết thành in hoa:',b)
print('-'*70)
#=====
#                               CỘNG CHUỖI
#=====

cong_chuoai = chuoai + chuoai_1 # cộng 2 chuỗi
print('Cộng hai chuỗi lại thành: ',cong_chuoai)
print('-'*70)
#=====
#                               ĐẾM SỐ KÝ TỰ TRONG CHUỖI
#=====

count = 0 # giá trị đếm ký tự bắt đầu từ 0
for i in chuoai: # i chạy hết chuỗi chuoai
    count += 1 # qua 1 ký tự là vòng for thì count tăng lên 1, và dừng vòng for khi hết chuỗi
print('Đây là số ký tự của chuỗi: ',count) # kết quả in ra cuối màn hình là số ký tự

for i in chuoai_1: # i chạy hết chuỗi chuoai
    count += 1 # qua 1 ký tự là vòng for thì count tăng lên 1, và dừng vòng for khi hết chuỗi
print('Đây là số ký tự của chuỗi 1: ',count) #
print('-'*70)
#=====
#                               ĐỊNH DẠNG CHUỖI
#=====

hoten = 'Tony Teo'
namsinh = '2000'
mssv = '1234567'
khoa = 'CNTT'

thongtin = 'Họ Tên:{}, Năm Sinh:{}, Mã Số Sinh Viên:{}, Khoa:{}' # sắp xếp theo thứ tự các chuỗi
print(thongtin.format(hoten,namsinh,mssv,khoa))
```

## D. BÀI TẬP



**Bài 1:** Cho phép người dùng nhập 2 chuỗi s1, s2. Xuất ra màn hình kết quả tổng của 2 chuỗi đó

**Gợi ý:**

- Dùng hàm input yêu cầu người dùng nhập giá trị
- Nhập chuỗi thứ nhất và gán vào biến s1
- Nhập chuỗi thứ hai và gán vào biến s2
- Xuất ra màn hình 2 chuỗi trên hoặc gán 2 chuỗi s1, s2 vào 1 chuỗi rồi xuất ra kết quả chuỗi cuối cùng

**Bài 2:** Viết hàm tìm xem chuỗi S1 có trong chuỗi S2 hay không?

**Ví dụ:**

- S1 = “viet nam dat nuoc toi yeu”;
- S2 = “dat nuoc”;
- Tìm xem chuỗi S2 có xuất hiện trong chuỗi S1 không?
- *Gợi ý: dùng hàm input yêu cầu người dùng nhập giá trị*
- *Xem từ khóa in, not in (kiểm tra chuỗi trong chuỗi)*

**Bài 3:** Cho trước chuỗi s1 = “Cong Hoa xa hoi chuu nghia Viet nam”

Hãy in HOA hết các từ trong chuỗi trên

In thường hết các từ trong chuỗi trên

Thay thế từ “chuu” thành “chu”

**Gợi ý:**

- Dùng hàm input yêu cầu người dùng nhập giá trị
- Dùng hàm upper để đổi chuỗi trên thành chữ HOA
- Dùng hàm lower để đổi chuỗi trên thành chữ thường
- Dùng hàm replace để thay thế chuỗi a thành chuỗi b

**Bài 4:** Giả sử có một chuỗi như sau: “0983876207;75;10:18:25;0918295063”, tách chuỗi trên thành từng phần riêng biệt.

Cho biết vị trí thứ 2 trong chuỗi trên là thời gian cuộc gọi từ số thứ 1 “0983876207” sang số thứ 2 “0918295063”. Biết rằng 1 phút cho mỗi cuộc gọi là 2500đ, tính giá cước cuộc gọi trên.





**Gợi ý:**

- Dùng hàm input yêu cầu người dùng nhập giá trị
- Dùng hàm split để tách chuỗi
- Xác định vị trí thời gian cuộc gọi là vị trí thứ 2
- Chuyển đổi số trên thành kiểu số (có khả năng tính toán được)
- 2500đ 1 phút thì tính ra 1s là bao nhiêu?
- Sau đó lấy 2 giá trị nhân nhau cho ra kết quả giá tiền của cuộc gọi điện thoại trên.

**Bài 5: HOW LONG IS A PIECE OF STRING?**

You need to design a program that will allow a user to input a string.

Take the user input and count the number of characters in the string.

Output the length of the string.

If the user was to input:

Have a nice day.

then the output should be as follows: The string is 16 characters long

**EXTENSION**

Add a choice to the program that will allow the user to select if they wish to count the spaces in the string along with the other characters.

Process the input string taking into account whether or not they wish to count the spaces.

**Bài 6: Python program to count words in a sentence**

Method #1: Using split() split function is quite useful and usually quite generic method to get words out of the list

**Output**

The original string is : Vanlang is best Computer Science Portal

The number of words in string are : 6

**Bài 7: THE GREAT ESCAPE**

This challenge looks at how we can escape strings in order to produce validated output.



When you work with other coders' code you will often find complex escaping patterns which can confuse at first sight.

It is therefore a good idea to be fairly comfortable with escaping strings.

### TASK

Your task is simply to take each of these sentences and output them in the browser.

`This is Sam's apple!`

`Sam said, "Hello, Colin. How's your new PHP course going?"`

`Sam's mother ordered a new book called "A Day in my Life" from her local bookstore.`

Replace the name Sam with a variable name and produce the required output

### Bài 8: Love Calculator

#### Instructions

You are going to write a program that tests the compatibility between two people.

To work out the love score between two people:

Take both people's names and check for the number of times the letters in the word TRUE occurs.

Then check for the number of times the letters in the word LOVE occurs.

Then combine these numbers to make a 2 digit number.