

BÀI 6: Numbers – Strings - Datetimes



Mục tiêu chính: Cung cấp cho HV kiến thức và kỹ năng sử dụng:

- Các hàm thư viện Numbers – Strings - Datetimes

Viết hàm max, min

- ✓ **Yêu cầu:** Viết lại bài tìm giá trị lớn nhất, nhỏ nhất bằng cách sử dụng hàm thư viện.
- ✓ **Hướng dẫn**
 - Trong project Python_co_ban, tạo package **Bai6**
 - Trong package Bai6, tạo module có tên là **tim_min_max_2.py**
 - Sử dụng hàm thư viện Numbers

6.2. Sử dụng hàm abs

- ✓ **Yêu cầu:** Viết lại bài tìm $|x|$ bằng cách sử dụng hàm thư viện.
- ✓ **Hướng dẫn**
 - Trong package Bai6, tạo module có tên là **tim_gttd_x_2.py**
 - Sử dụng hàm thư viện Numbers

6.3. Sử dụng hàm pow

- ✓ **Yêu cầu:** Viết lại bài tính $S = (x^2 + 1)^n$ bằng cách sử dụng hàm thư viện.
- ✓ **Hướng dẫn**
 - Trong package Bai6, tạo module có tên là **tinh_S_2.py**
 - Sử dụng hàm thư viện Numbers

6.4. Sử dụng hàm pow

- ✓ **Yêu cầu:** Viết lại bài tính : $A = (x^2 + x + 1)^n + (x^2 - x + 1)^n$ bằng cách sử dụng hàm thư viện.
- ✓ **Hướng dẫn**
 - Trong package Bai6, tạo module có tên là **tinh_A_2.py**
 - Sử dụng hàm thư viện Numbers

6.5. Kiểm tra dữ liệu zip code Việt Nam

- ✓ **Yêu cầu:** Bạn hãy viết một phương thức giúp kiểm tra dữ liệu mã zip (zip code) của Việt Nam:
 - ✓ Với một chuỗi truyền vào, sẽ kiểm tra chuỗi đó có là một mã zip hợp lệ hay không? Nếu hợp lệ thì trả về True, ngược lại thì trả về
 - ✓ Mã zip hợp lệ là bất kỳ chuỗi nào bao gồm chính xác 6 chữ số.

6.6. Giải phương trình bậc 2



✓ **Yêu cầu: Xây dựng chương trình giải phương trình bậc 2: $ax^2 + bx + c = 0$**

- Sử dụng shell
- Nhập vào a, b, c => Giải phương trình và in kết quả

Giải phương trình bậc 2: $ax^2 + bx + c = 0$

Nhập a:

2

Nhập b:

-7

Nhập c:

3

Phương trình bậc 2:

Phương trình có hai nghiệm phân biệt $x_1 = 3.0$, $x_2 = 0.5$

Giải phương trình bậc 2: $ax^2 + bx + c = 0$

Nhập a:

6

Nhập b:

1

Nhập c:

5

Phương trình bậc 2:

Phương trình vô nghiệm

Giải phương trình bậc 2: $ax^2 + bx + c = 0$

Nhập a:

1

Nhập b:

-8

Nhập c:

16

Phương trình bậc 2:

Phương trình có nghiệm kép $x_1 = x_2 = 4.0$



Giải phương trình bậc 2: $ax^2 + bx + c = 0$

Nhập a:

0

Nhập b:

2

Nhập c:

4

Phương trình bậc 2 suy biến thành phương trình bậc 1:

Nghiệm = -2.0

✓ **Hướng dẫn sử dụng:**

- Nhập vào a, b, c => Hiển thị nghiệm của phương trình

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- a, b, c

▪ **Xuất:**

- Nghiệm x1, x2

▪ **Qui tắc xử lý :**

- Nếu a = 0: giải phương trình bậc I $bx+c = 0$
 - Nếu b = 0, c <> 0: phương trình vô nghiệm
 - Nếu b = 0, c = 0: phương trình vô số nghiệm
 - Nếu b <> 0: phương trình có nghiệm $x = -c/b$
- Nếu a <> 0:
 - Tính $\Delta = b^2 - 4ac$
 - Nếu $\Delta < 0$ thì phương trình vô nghiệm.
 - Nếu $\Delta = 0$ thì phương trình (2) có nghiệm kép : $x_1 = x_2 = x_0 = \frac{-b}{2a}$
 - Nếu $\Delta > 0$ thì phương trình (2) có 2 nghiệm phân biệt : $x_{1,2} = \frac{-b \pm \sqrt{\Delta}}{2a}$

✓ **Hướng dẫn**

- Trong package Bai6, tạo module có tên là **giai_PT_bac_2.py**
 - Áp dụng quy tắc xử lý trên
 - Sử dụng hàm thư viện Numbers

6.7. Tìm kiếm từ (keyword) trong danh sách các chuỗi

✓ **Yêu cầu:**

- Một nhà nghiên cứu đã thu thập hàng nghìn bài báo, nhưng cô ấy muốn tập trung sự chú ý của mình vào các bài báo bao gồm một từ cụ thể.



- Bạn hãy viết phương thức ***word_search(doc_list, keyword)*** để giúp cô ấy lọc danh sách các bài báo của mình, với giá trị trả về là danh sách các giá trị chỉ mục vào danh sách ban đầu cho tất cả các tài liệu có chứa từ khóa.

```
doc_list = ["The Learn Python Challenge Casino.", "They bought a car", "Casinoville"]
>>> word_search(doc_list, 'casino')
>>> [0]
```

- Phương thức của bạn phải đáp ứng các tiêu chí sau:
 - Không bao gồm các tài liệu mà chuỗi từ khóa chỉ hiển thị như một phần của từ lớn hơn. Ví dụ, nếu cô ấy đang tìm kiếm từ khóa "closed", thì sẽ không bao gồm chuỗi "enclosed".
 - Cô ấy không muốn bạn phân biệt chữ hoa và chữ thường. Ví dụ, cụm từ "Closed the case." sẽ được đưa vào khi dùng từ khóa "closed".
 - Đừng để dấu chấm hoặc dấu phẩy ảnh hưởng đến tìm kiếm. Ví dụ, "It is closed." sẽ được đưa vào khi dùng từ khóa "closed".

✓ **Hướng dẫn:**

- Nhập vào danh sách các chuỗi → biến doc_list
- Viết phương thức word_search:
 - Làm sạch các chuỗi: cắt bỏ khoảng trắng, dấu ".", ",", ... ở hai đầu chuỗi, đổi sang chữ thường.
 - Vòng lặp duyệt qua mỗi chuỗi và ứng với mỗi chuỗi thì tìm kiếm keyword trong chuỗi. Nếu tìm thấy thì lưu lại vị trí tìm thấy.

6.8. Tìm kiếm nhiều từ (multi keyword) trong danh sách các chuỗi

✓ **Yêu cầu:**

- Bây giờ nhà nghiên cứu muốn cung cấp nhiều từ khóa để tìm kiếm.
- Dựa vào bài tập trên, bạn hãy viết phương thức ***multi_word_search(doc_list, keywords)*** để giúp cô ấy tìm các bài báo mà cô ấy cần, với giá trị trả về là một từ điển (dictionary), trong đó mỗi khóa là một từ khóa và giá trị là danh sách các chỉ mục (từ doc_list) của các tài liệu có chứa từ khóa đó.

```
doc_list = ["The Learn Python Challenge Casino.", "They bought a car and a casino", "Casinoville"]
keywords = ['casino', 'they']
>>> multi_word_search(doc_list, keywords)
>>> {'casino': [0, 1], 'they': [1]}
```

6.9. Phân tích của Luigi về các vật phẩm để giành chiến thắng trong các cuộc đua

✓ **Yêu cầu:**

- Luigi đang cố gắng thực hiện một phân tích để xác định các vật phẩm tốt nhất để giành chiến thắng trong các cuộc đua trên đường đua Mario Kart. Anh ta có một số dữ liệu dưới dạng dictionary như sau:

```
[
    {'name': 'Peach', 'items': ['green shell', 'banana', 'green shell'], 'finish': 3},
```



```
{'name': 'Bowser', 'items': ['green shell'], 'finish': 1},
# Đôi khi tên của tay đua không được ghi
{'name': None, 'items': ['mushroom'], 'finish': 2},
{'name': 'Toad', 'items': ['green shell', 'mushroom'], 'finish': 1},
]
```

- 'item' là danh sách tất cả các vật phẩm tăng sức mạnh mà tay đua nhặt được trong cuộc đua đó, và 'kết thúc' là vị trí của chúng trong cuộc đua (1 cho vị trí đầu tiên, 3 cho vị trí thứ ba, v.v.).
- Hãy viết phương thức **best_items(racers)** để giúp anh ấy lấy một danh sách đã cho, và trả về một dictionary ánh xạ về số 'item' được chọn bởi những người về đích đầu tiên ('finish':1).

✓ **Hướng dẫn:**

- Nhập vào danh sách các tay đua

```
sample = [
    {'name': 'Peach', 'items': ['green shell', 'banana', 'green shell'], 'finish': 3},
    {'name': 'Bowser', 'items': ['green shell'], 'finish': 1},
    {'name': None, 'items': ['mushroom'], 'finish': 2},
    {'name': 'Toad', 'items': ['green shell', 'mushroom'], 'finish': 1},
]
```

- Viết phương thức **best_items(racers)**
- Xuất kết quả:

```
print(best_items(sample))

{1: 1, 3: 2}
```

6.10. Sử dụng hàm Strings

✓ **Yêu cầu: Xây dựng chương trình xử lý chuỗi**

- Sử dụng shell
- Nhập vào chuỗi s, chuỗi s_sub, s_find, s_replace.
 - In chuỗi s
 - Loại bỏ khoảng trắng ở đầu và cuối chuỗi
 - In chuỗi với ký tự đầu chuỗi viết hoa
 - Đếm và in ra số lần chuỗi con s_sub xuất hiện trong chuỗi s
 - Tìm kiếm s_find trong s và thay thế bằng s_replace, in chuỗi sau khi tìm kiếm và thay thế

```

Nhập chuỗi s:
a b c d e f duck
Nhập chuỗi con s_sub:
d
Nhập chuỗi tìm s_find:
duck
Nhập chuỗi thay thế s_replace:
dog
Chuỗi s sau khi loại bỏ khoảng trắng đầu và cuối chuỗi: a b c d e f duck
Chuỗi viết hoa ký tự đầu: A b c d e f duck
Số lần s_sub xuất hiện trong s: 2
Chuỗi s sau khi tìm kiếm và thay thế: A b c d e f dog
    
```

✓ **Hướng dẫn sử dụng:**

- Nhập vào s, s_sub, s_find, s_replace => Hiển thị kết quả như hình trên

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- s, s_sub, s_find, s_replace

▪ **Xuất:**

- s

✓ **Hướng dẫn**

- Trong package Bai6, tạo module có tên là **xu_ly_chuoi.py**
 - Sử dụng hàm thư viện Strings

6.11. Sử dụng hàm Datetimes

✓ **Yêu cầu: Xây dựng chương trình xử lý ngày, tháng, năm**

- Sử dụng shell
- Nhập vào ngày, tháng, năm (hợp lệ)
 - Xuất ngày theo định dạng ngày – tháng - năm
 - Cho biết năm được nhập vào có phải là năm nhuận hay không?
 - Cho biết ngày/tháng/năm nhập vào là thứ mấy?
 - Cho biết tháng nhập vào có bao nhiêu ngày?



```

Nhập ngày:
18
Nhập tháng:
2
Nhập năm:
2017
Ngày tháng năm vừa nhập: 18 - 02 - 2017
Năm 2017 không là năm nhuận
18 - 02 - 2017 là Thứ Bảy
Số ngày trong tháng 2 là: 28
    
```

- ✓ **Hướng dẫn sử dụng:**
 - Nhập vào ngày, tháng, năm => Hiển thị kết quả như hình trên
- ✓ **Tóm tắt yêu cầu**
 - **Nhập:**
 - Ngày, tháng, năm
 - **Xuất:**
 - Thông tin theo yêu cầu
- ✓ **Hướng dẫn**
 - Trong package Bai6, tạo module có tên là **xu_ly_thoi_gian.py**
 - Sử dụng hàm thư viện Datetimes