**Bài tập python**

<https://www.practicepython.org/exercises/>

**Bài 1. Những kiến thức cơ bản python (kiểu dữ liệu, các cấu trúc điều khiển)**

**Bài 1.1** Tạo một chương trình yêu cầu người dùng nhập tên và tuổi của họ. In ra một thông báo cho họ nói với họ năm mà họ sẽ tròn 100 tuổi.

**Mở rộng:** Thêm vào chương trình đã viết bằng cách yêu cầu người dùng cho một số và in ra số bản sao thông báo bằng số người dùng nhập vào.

In ra nhiều bản sao của thông báo trước đó trên các dòng riêng biệt. (Gợi ý: chuỗi "\ n giống như nhấn nút ENTER)

**Bài 1.2** Yêu cầu người dùng cho một số, tùy thuộc vào số đó là số chẵn hay số lẻ, hãy in ra một thông điệp phù hợp cho người dùng.

**Mở rộng:** Nếu số đó là bội số của 4, hãy in ra một tin nhắn khác.

Yêu cầu người dùng cho hai số: một số để kiểm tra và một số để chia cho (kiểm tra). Nếu kiểm tra chia hết cho số kiểm tra thì hãy thông báo điều đó cho người dùng. Nếu không, in một thông báo thích hợp khác.

**Bài 1.3.** Cho một danh sách, ví dụ như sau: a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]

và viết một chương trình in ra tất cả các phần tử của danh sách nhỏ hơn 5.

**Mở rộng**: Thay vì in từng phần tử một, hãy tạo một danh sách mới có tất cả các phần tử nhỏ hơn 5 từ danh sách đó, viết nó trong một dòng.

Yêu cầu người dùng nhập vào một số và trả về một danh sách chỉ chứa các phần tử từ danh sách *a* nhỏ hơn số đó.

**Bài 1.4.** Cho hai danh sách, ví dụ hai cái này:

   a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]

   b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]

và viết chương trình trả về một danh sách chỉ chứa các phần tử chung giữa các danh sách (không trùng lặp).

**Mở rộng**: Tạo ngẫu nhiên hai danh sách và thực hiện yêu cầu trên (ví dụ a = *random.sample(range(100), 5))*

**Bài 1.5** Cho một danh sách được lưu trong biến: *a* = [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]. Viết một dòng lệnh băng ngôn ngữ Python tạo ra một danh sách mới từ *a* chỉ có các phần tử chẵn.

**Tương tự cho:** cho danh sách năm sinh years\_of\_birth = [1990, 1991, 1990, 1990, 1992, 1991] hãy viết 1 dòng lệnh tính tuổi đến năm 2020 của những người có năm sinh trong danh sách.

Bài 2. Lập trình hướng thủ tục (function)

<https://www.w3resource.com/python-exercises/python-functions-exercises.php>

**Bài tập 1**. Đoc ghi file txt

* Tạo file text tên **quatmo.txt** có nội dung như sau:

*Thang Bom co cai quat mo*

*Phu ong xin doi ba bo chin trau*

*Bom rang Bom chang lay trau*

*Phu ong xin doi ao sau ca me*

Viết chương trình thực hiện: Đọc file dữ liệu trên và in lên màn hình, sau đó yêu cầu người dùng nhập thêm các câu thơ của bài thơ vào file quatmo.txt cho đến khi người dùng nhập vào chữ “end” thì kết thúc

*Bom rang Bom chang lay me*

*Phu ong xin doi ba be go lim*

*…*

**Bài 2. Tạo thư mục, tao file**

Viết chương trình tạo ra một thư mục “ABC” trong thư mục hiện hành và tạo thêm 1 file **text1.txt** trong thu mục “ABC”. Yêu cầu người dùng nhập vào tên thư mục XYZ. Đổi tên thư mục ABC thành XYZ, sau đó đưa ra thông báo bạn có muốn xóa file text1.txt không (Yes/No)? Nếu người dùng chọn Yes thì xóa file này đi.

**Bài 2. Xâu ký tự**

Cho tập tin văn bản *dssv.txt* chứa danh sánh họ và tên sinh viên của một lớp:

*Nguyen van nam*

*Ha tuan anh*

*NGUYEN tran tuan*

*le nam thanh*

*nguyen tuan ANH*

Hãy viết chương trình thực hiện các công việc sau:

1. Chuẩn hóa họ tên của danh sách sinh viên trong tệp tin dssv.txt, sắp tên của các sinh viên theo thứ tự *abc* sau đó ghi ra tệp tin mới với tên dssv\_2.txt

*Chú ý: họ tên chuẩn hóa là các từ của tên chỉ viết hoa chữ cái đầu tiên*)

1. Thống kê các thông số sau đây:
   1. Có bao nhiêu sinh viên họ “Nguyen”
   2. Có bao nhiêu sinh viên tên “Anh”

Bài 3. Lập trình hướng đối tượng

<https://www.w3resource.com/python-exercises/class-exercises/index.php>

**Bài tập 1.**

**-** Xây dựng lớp ConVat gồm các thuộc tính: Họ (mèo, linh trưởng, rùa…), Năm sinh, Môi trường sống (Trên cạn, Dưới nước, Trong đất,..); các phương thức: Nhập thông tin con vật, in thông tin con vật lên màn hình (\_\_str\_\_).

- Xây dựng lớp “ThuCung“ (thú cưng) thừa kế từ lớp ConVat, bổ sung thêm các thuộc tính: Mã, Tên, Giống, viết lại phương thức nhập thông tin con vật, in thông tin con vật (\_\_str\_\_).

- Xây dựng lớp quản lý danh sách con vật có các phương thức: thêm con vật, xóa con vật khi biết mã, thống kê số lượng con vật theo từng họ, nhập danh sách các con vật từ file, ghi danh sách các con vật ra file theo cấu trúc sau:

STT, Ma, Ten, Giong, Ho, Nam sinh, Moi truong

1, 123, John, Fox, Dog, 2018, Can

2, 124, Mix, Bec, Dog, 2017, Can

3 125, Kissty , English, Cat, 2019, Can

**Bài tập 2.** Để quản lý hồ sơ học sinh của trường THPT nhà trường cần các thông tin sau: Mã, tên lớp, họ tên, ngày sinh, giới tính

**Yêu cầu 1:** Xây dựng lớp **HocSinh** để quản lý thông tin của mỗi học sinh, lớp này có phương thức nhập thông tin của học sinh từ bàn phím, phương thức in thông tin của học sinh trên một dòng.

**Yêu cầu 2:** Xây dựng lớp QuanLyHocSinh để quả lý danh sách học sinh của toàn trường có các phương thức:

+ Thêm danh sách học sinh cho đến khi người dùng vào mã học sinh là “end”

+ Ghi danh sách sinh viên ra tện “**dshs.txt**”

+ Đọc danh sách học sinh vào chương trình từ tệp “**dshs.txt**”

+ Đếm tổng số học sinh nữ/ hoặc nam khi người dùng cung cấp thông tin giới tính.

+ Thống kê số học sinh của từng lớp và ghi ra file **sisolop.txt** theo cấu trúc:

*Tên lớp Sĩ số*

*1A1 60*

*1A2 55*

**Bài 4. Thư viện Numpy**

<https://numpy.org/devdocs/user/quickstart.html#linear-algebra>

<https://www.w3resource.com/python-exercises/numpy/index.php>

**Bài tập 1.**

* Tạo mảng numpy một chiều từ dãy số sau đây {1, 4, 3, 7, 2, 5, 3, 3} hoặc tạo ra bằng các hàm arange với số phần tử tùy ý.
* Từ mảng 1 chiều ở trên tạo ra một mảng 2 chiều có **2 hàng** số cột tùy ý
* Từ mảng 2 chiều ở trên chuyển về mảng 2 chiều có 3 hàng
* Từ mảng 2 chiều ở trên tạo ra mảng 1 một chiều
* Tạo một mảng 3 chiều ngẫu nhiên có kích thước 3x5x6

**Bài tập 2.**

Cho ma trận A =

1. Tìm ma trận nghịch đảo của ma trận A
2. Tìm ma trận chuyển vị của ma trận A
3. Tìm ma trận tích A × A
4. Cho vector Y = [3, 2, 4] hãy giải phương trình A × X = YT

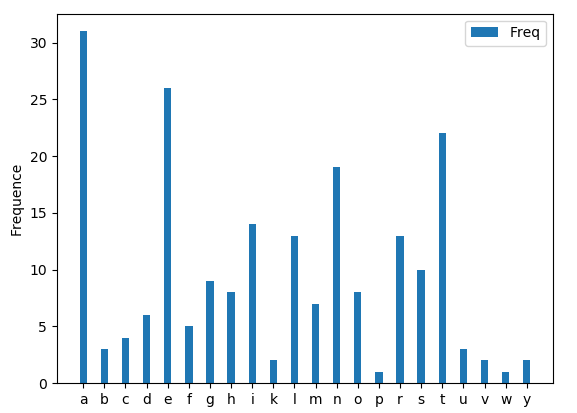
**Bài 5. Thư viện Mathplotlib**

[https://matplotlib.org/3.1.1/gallery/index.html#](https://matplotlib.org/3.1.1/gallery/index.html)

**Bài tập 1**. Vẽ đồ thị các hàm số dưới đây trong đoạn [-50, 50]

**Bài tập 2**: Viết chương trình thực hiện yêu cầu sau đây:

1. Đọc vào nội dung của một tập tin văn bản tiếng Anh. Thống kê số chữ cái trong văn bản và tần xuất suất hiện của các chữ cái này (không phân biệt chữ hoa và chữ thường).
2. Vẽ biểu đồ dạng cột biểu thị tần suất của các chữ cái xuất hiện trong văn bản như hình dưới đây



**Bài tập 3**: Viết chương trình thực hiện yêu cầu sau đây:

1. Đọc vào nội dung của một tập tin văn bản tiếng Anh trong một file txt (văn bản chỉ chứa các từ, không chứa ký tự đặc biệt và hình vẽ). Thống kê số từ khác nhau có trong văn bản và tần xuất suất hiện của mỗi từ trong văn bản.
2. Vẽ biểu đồ dạng cột biểu thị tần suất của **10 từ có tần xuất cao nhất** trong văn bản

Ví dụ file **news.txt** có nội dung như sau

English is a West Germanic language that was first spoken in early medieval England and eventually became a global lingua franca.  
It is named after the Angles, one of the Germanic tribes that migrated to the area of Great Britain that later took their name, as England.

**Bài 6. Một số thư viện chuẩn của python**

**Bài 1.** Viết chương trình tạo ra 1 mảng numpy có 1000 số nguyên trong khoảng [0,10000]. Viết hàm sắp xếp bằng thuật toán quick sort và hàm sắp xếp bằng thuật toán nổi bọt (bubble sort). Thực hiện sắp xếp dãy số trên bằng 2 thuật toán và so sánh thời gian chạy của 2 thuật toán đó trên dãy số đã tạo.

**Bài 2.** Cho dãy giá trị x = [3, 23, 23, 32, 23,12, 83, 42, 92] và y = [5, 23, 32, 52, 233, 84, 83, 432, 192] hãy tạo dãy tọa độ (xi, yi) từ 2 dãy số x, y; sắp dãy tọa độ theo thứ tự tăng dần của yi.

**Bài 3.** Viết chương trình quản lý danh sách nhân viên, mỗi nhân viên cần quản lý các thông tin: Họ tên, tuổi, giới tính (name, nữ, khác), địa chỉ, hệ số lương. Chương trình có các chức năng sau:

* Nhập danh sách nhân viên từ bàn phím
* Nhập danh sách nhân viên từ file
* Ghi danh sách nhân viên ra file
* Sắp xếp danh sách nhân viên theo tuổi bằng hàm sorted hoặc phương thức sort của list

**Bài 4.** Viết chương trình yêu cầu người dùng nhập vào đường dẫn tới **một thư mục**, thực hiện kiểm tra:

* Nếu thư mục tồn tại thì tính dung lượng của các file trong một thư đó, in lên màn hình thời gian tính toán của chương trình.
* Nếu thư mục không tồn tại thì thông báo “Thư mục không tồn tại”.

**Bài 5.** Viết chương trình yêu cầu người dùng nhập vào đường dẫn tới **một thư mục**, thực hiện kiểm tra:

* Nếu thư mục tồn tại thì **xóa** thư đó đi, in lên màn hình thời gian thực hiện xóa thư mục đó.
* Nếu thư mục không tồn tại thì thông báo “Thư mục không tồn tại”.

**Bài 6.** Viết chương trình vẽ đồ thị hàm số sau đây:

Trong đó *a, b, c* là các tham số được đưa vào khi chạy chương trình. (*Ví dụ: python vedthi.py –a 10 -b 5 –c* 3)

**Bài 7. Phát triển webservice với python**

<https://blog.miguelgrinberg.com/post/designing-a-restful-api-with-python-and-flask>

**Bài 7. Tìm hiểu thư viện học máy (tensorflow)**

<https://www.tensorflow.org/>

<https://github.com/tensorflow/tensorboard>

<https://nguyenvanhieu.vn/khoa-hoc-tensorflow/>

**Bài 8. Tìm hiểu lập trình học máy trên**

<https://nttuan8.com/bai-1-linear-regression-va-gradient-descent/#Python_code>

[https://colab.research.google.com](https://colab.research.google.com/notebooks/welcome.ipynb#scrollTo=K0JlQqkQ7u_w)

<https://forum.machinelearningcoban.com/t/su-dung-google-colab-voi-tesla-k80-gpu-mien-phi/2275>

**Trainning Model sử dụng Google Colab**

Bước 1:

Tạo Notebook:

Vào drive, tạo 1 thư mục mang tên gì đó, (VD: AI\_COLAB ), vào thư mục

- Chuột phải, chọn “ứng dụng khác” (more) -> “kết nối ứng dụng khác” . Cửa sổ hiện lên, gõ keyword “Colab” và cài luôn extension đầu tiên

- Sau khi cài xong, quay lại thư mục -> chuột phải -> “ứng dụng khác” (more) -> “kết nối ứng dụng khác” -> Colaboratory

Bước 2:

Kết nối Drive:

* Không bắt buộc nhưng cần thiết để chứa train data, test data, các lib thêm khác
* run:

!apt-get update -qq 2>&1 > /dev/null

!apt-get install -y -qq software-properties-common python-software-properties module-init-tools

!add-apt-repository -y ppa:alessandro-strada/ppa 2>&1 > /dev/null

!apt-get update -qq 2>&1 > /dev/null

!apt-get -y install -qq google-drive-ocamlfuse fuse

* Tiếp:

from google.colab import drive

drive.mount('/content/drive')

* output hiện ra 1 đường link, click vào link đó, đăng nhập google, copy token hiện thị ra và paster vào ô input trên notebook, như vậy bạn đã cấp quyền cho google colab được phép truy cập vào drive của bạn
* Trong của sổ bên trái, click vào Files -> REFRESH , bạn sẽ thấy Drive của bạn đã được thêm vào working directory

Bước 3: Run code

**Run source code từ Github:**

Bước 1:

Clone code:

!git clone “*path(http:www. ...)*”

Bước 2:

Run code:

!python “*path\_to\_main\_file*”

**Run source code từ Google Drive:**

Bước 1:

Kết nối Drive như bước 2 phần đầu tiên

Bước 2:

Run code

Google cung cấp thời gian chạy cho GPU (Tesla K80 ) là 8~12 tiếng. Có thể bị ngắt sớm hơn nếu như tải quá nặng hoặc mạng kém.

Lưu trữ lại mô hình để training tiếp trong trường hợp bị out, và tiếp tục training mô hình vừa lưu lại được (<https://medium.com/@prajwal.prashanth22/google-colab-drive-as-persistent-storage-for-long-training-runs-cb82bc1d5b71>)

Ví dụ: Có một source code lưu trên máy cần được trainning, các bước làm như sau:

Bước 1:

Đưa source code về GG Drive bằng cách:

* Source code trên máy: tạo upload folder lên gg drive

Bước 2:

Mount Drive: gõ lệnh

*from google.colab import drive*

*drive.mount('/content/drive')*

nhấn vào link hiện ra và copy code vào ô trống và nhấn enter

nhấn refresh vào thanh Navigation bên trái trong mục files để xem danh mục file đã hiện ra folder drive chưa

Kiểm tra thư mục hiện tại con trỏ đang đứng bằng lệnh:

*!ls*

và thay đổi con trỏ đến vị trí mới bằng lệnh:

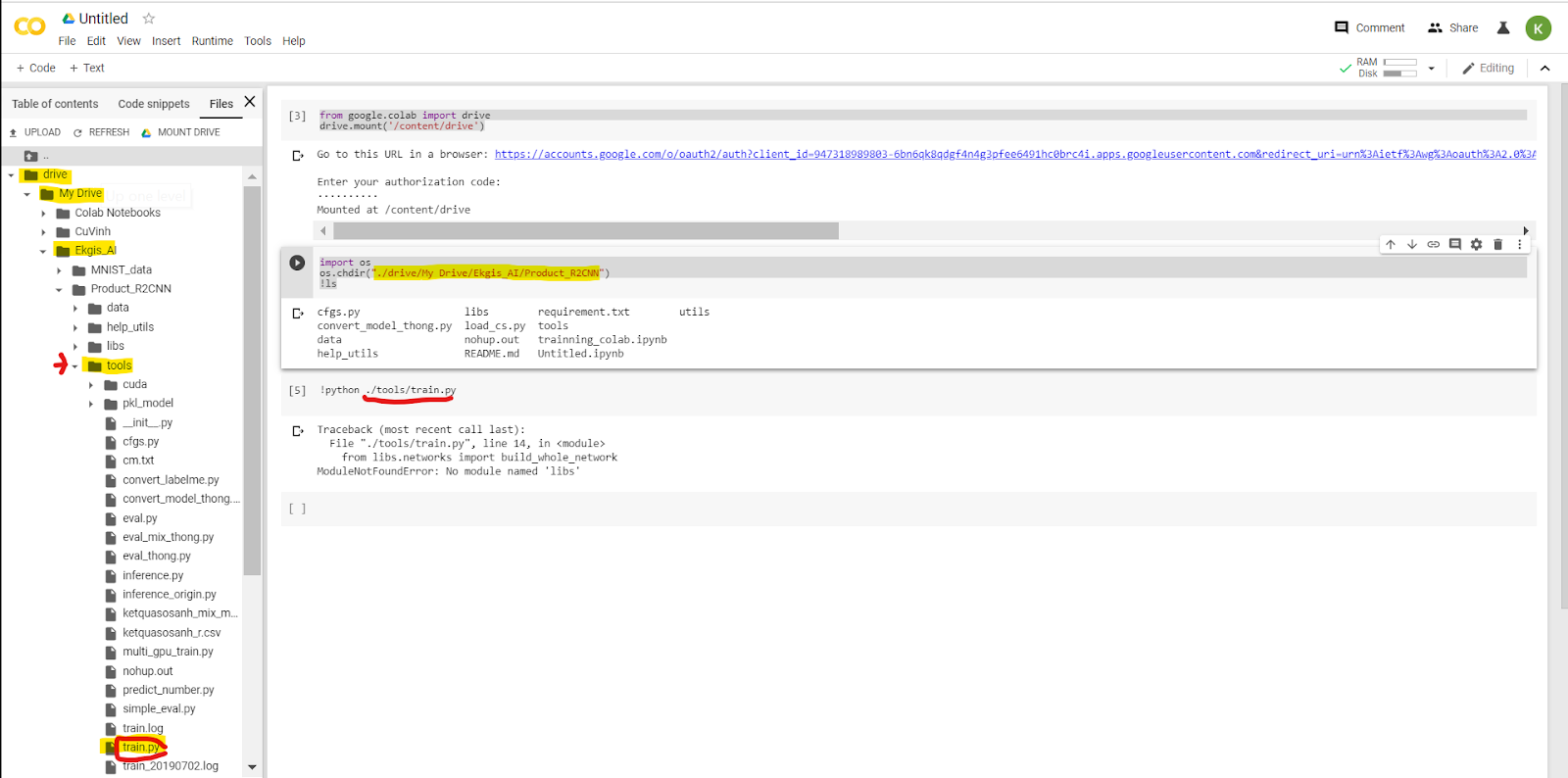
*import os*

*os.chdir(“path\_to\_directory”)*

**Bước 3:**

**run code:**

*!python “path\_to\_file”*

****