

## I. General knowledge

1. `dir`: List các `file` trong folder
2. Dấu `"=`": Assign giá trị  
`"=="`: Boolean function  
`"!="`: Different
3. `Parse` = convert
4. `for i in range(start, stop, step):`  
`start`: default = `0`; `stop`: default = `n - 1`; `step`: default = `1`  
`i`: loop counter
5. `length += 5` (`length = length + 5`)
6. Lưu thông tin: Lưu biến
7. Lowercase: `lower()`
8. Uppercase: `upper()`
9. Shuffle

10. Hàm `randint` và `choice`  
`from random import randint`  
`random_number = randint(0, 100)`

`from random import choice`  
`random_item = choice(list)`

11. Lưu string trên nhiều dòng

```
string = """
My name is
Nam
"""
```

12. Database structure in NoSQL

Database - Collection - Document  
Collection: managed by database user

## II. Các lệnh trong List

1. Delete  
`remove`  
`del list[i]`  
`pop`

2. Break: phá vỡ vòng lặp gần nhất. Nếu dùng biến để dừng vòng lặp thì dừng ở lần tiếp theo

3. Đầu `list`: `0`  
Cuối `list`: `-1`

4. Lặp từ chỗ nào ◊ cho vào vòng lặp từ chỗ đấy

5. Kiểm tra `list` rỗng hay không: `len(list) == 0`

6. Create `list`

```
List = [ , , , ]
```

7. Print

```
Print(*list)
```

8. Check length of `list`

```
len(list)
```

9. Check value trong `list`

```
List[i] = ...
```

10. Print `list`

```
for i in range(5):  
    print(i + 1, ".", list[i], sep = " ")
```

```
for i in range(5):  
    print("{0}. {1}".format(i + 1, list[i]))
```

```
for index, item in enumerate(list):  
    print("{0}. {1}".format(i + 1, item))
```

```
for item in list:  
    print(item)
```

11. Update `list`

Replace:

```
list[i] = replace_item
```

Add:

```
List.append(content)
```

Check position:

```
Position = list.index("value")
```

12. Tách phần tử trong string

```
list(string)
```

13. Tách `list` by space

```
list.strip().split(sep = " ")
```

14. List có thể chứa integer, string, các loại khác. Chỉ nên lưu dữ liệu cùng loại, không nên lưu dữ liệu loại khác

```
15. Replace trong string
string = "Sorry I was {{sickness}}"
new_string = string.replace("{{sickness}}", "under the weather")
```

### III. Các lệnh trong dictionary

Dictionary: 1 cặp key và value; ngăn cách nhau bởi dấu “,”

Phần tử trong dictionary có thể coi như 1 biến

#### 1. Khai báo dictionary

```
dictionary = {
    "a": 1,
    "b": 2,
    "c": 3
}
```

#### 2. Kiểm tra phần tử có trong list hay không:

```
if item in list:
```

#### 3. Thêm mới vào dictionary:

```
dictionary[key] = value
```

Update trong dictionary:

```
dictionary[key] = new_value
```

Thêm mới khác với update: có key mới

#### 4. Đặt biến theo value trong dictionary

```
variable = dictionary[key]
```

#### 5. Print

```
print(dictionary)
```

#### 6. Delete

```
del dictionary[key]
```

#### 7. Kiểm tra phần tử có trong dictionary không

```
if key in dictionary:
```

```
    print("key")
```

```
else:
```

```
    print("not found")
```

#### 8. Print item in list

```
for k in dictionary:
```

```
    print(k, dictionary[k])
```

```
for key, value in dictionary.items():
```

```
    print(key, value)
```

```
for value in dictionary.values():
```

```
print(value)
```

9. Print cách 1 line: bấm enter để tiếp tục

```
print()
```

```
*****
```

Example 1. Check whether a number is a prime number

```
number = int(input("Please input number"))
```

```
is_prime = True
```

```
# Process
```

```
for i in range(2,number):
```

```
    if number % i == 0 :
```

```
        is_prime = False
```

```
        break
```

```
# Output
```

```
if is_prime == True:
```

```
    print("{0} is a prime number".format(number))
```

```
else:
```

```
    print("{0} is not a prime number".format(number))
```

Example 2. Check whether a list is sorted or not. If sorted, print "sorted".

If not, sort the list

```
enter_list = input("Please enter your list, separated by space")
```

```
new_list = enter_list.strip().split(sep=" ")
```

```
# List ko ro sort chua
```

```
numbers = []
```

```
# final list
```

```
sorted_list = []
```

```
length = len(new_list)
```

```
# chuyen list sang int
```

```
for item in new_list:
```

```
    numbers.append(int(item))
```

```
# Solution
```

```
print(numbers)
```

```
is_sorted = True
```

```
for i in range(len(numbers)-1):
```

```
    if numbers[i] > numbers[i+1]:
```

```
        is_sorted = False
```

```
        break
```

```
if is_sorted:
```

```
    print("Your sequence is sorted")
```

```

else:
    print("Your sequence is not sorted")
    print()
    for j in range(0,len(numbers)):
        min_number = min(numbers)
        position = numbers.index(min_number)
        del numbers[position]
        sorted_list.append(min_number)

    print("After sorted: ",sorted_list)

```

Example 3. Login 3 times

*# Solution*

```

print("Hi there is a superuser gateway")
count = 0

while True:
    user_name = input("Username")
    if user_name == "c4e":
        password = input("Password: ")
        if password == "codethechange":
            print("welcome")
            break
        else:
            print("password incorrect")
    else:
        print("You are not a superuser")
count += 1
if count == 3:
    print("You failed to login 3 times, go away")
    break

```

Example 4. Control the while loop

Count = 0

Loop = True

```

while loop:
    if count >= 7:
        loop = False
    else:
        Run the program
    count += 1

```

```

while loop:
    print("Running")
    count += 1
    if count ==5:
        loop = False

```

Example 5.

```
n = int(input("Please input the number"))
for i in range(n):
    print(n - i)
```

Example 6.

```
for i in range(3):
    for j in range(5):
        print("* ", end = " ")
    print()
```

Example 7. Basic sending email using (pip install gmail)

```
from gmail import GMail
from gmail import Message
from random import choice

html_content = """
    abcxyz
    """

placeholder: thay noi dung trong content

list_nghi = ["dau bung", "dau dau", "cam cum"]
reason = choice(list_nghi)
new_content = html_content.replace("{{sickness}}",reason)

gmail = GMail("daohoangnam@gmail.com","thisnoexist")
msg = Message("Test message 1",
    to = "emailtosend@gmail.com",
    html = new_content)

gmail.send(msg)
```

## IV. Matplotlib in Python

```
import matplotlib
matplotlib.use("TkAgg")
from matplotlib import pyplot
```

1. Prepare data

```
labels = ["iOS", "Android", "Web", "Reactive native"]
values = [20, 15, 40, 25]
colors = ["red", "blue", "black", "yellow"]
explode = [ 0, 0, 0, 0.2]
```

2. Draw (plot)

```
pyplot.pie(values, Labels = labels, colors = colors, explode = explode, shadow = True)
pyplot.axis("equal")
```

3. Show

```
pyplot.show()
```

## V. Extract website data

1. Download webpage

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import pyexcel
```

1.1. Create a connection

```
url = "http://dantri.com.vn"
connection = urlopen(url)
```

1.2. Read

```
data = connection.read()
print(data)
```

1.3. decode

```
html_content = data.decode("utf-8")
```

(bạn viết tắt ngắn gọn (from 1.1 to 1.3):

```
html_content = urlopen(url).read().decode("utf-8") )
```

1.4. Save html content to file (google cụm từ input - output)

```
f = open("./dantri.html", "wb")
f.write(html_content)
f.close()
```

2. Extract ROI (region of interest)

2.1. Chuyển định dạng về soup. BeautifulSoup có thể phân tích nhiều định dạng (html, xhtml, xml)

```
soup = BeautifulSoup(html_content, "html.parser")
```

2.2. làm đẹp html (tab lại cho đẹp)

```
print(soup.prettify())
```

2.3. Tìm bắt súp (trong trường hợp này là thẻ ul)

(tìm thẻ ul, class = "ul1 ulnew")

sử dụng: find/find\_all: Tên thẻ - đặc điểm (class)

Nếu không viết định danh attribute thì mặc định là class (xem trong documentation ở crummy.com)

```
ul = soup.find("ul", class = "ul1 ulnew")
```

#### 2.4. Tìm tiếp ROI

```
li_list = ul.find_all("li")
```

```
list_1 = []
```

```
for li in li_list:  
    item = {}
```

Có 3 cách tìm:

C1:

```
h4 = li.find("h4")  
a = h4.find("a")
```

C2:

```
h4 = li.h4  
a = h4.a
```

C3:

```
a = li.h4.a
```

```
title = a.string  
link = url + a["href"]  
# print(a.string)  
# print(url + a["href"])  
item['value'] = link  
item["link"] = title  
list_1.append(item)
```

#### 3. Extract info

```
pyexcel.save_as(records= dictionary, dest_file_name="link.xlsx")
```

## VI. Basic database example

```
pip install pymongo
```

```
pip install gmail
```

```
https://api.mongodb.com/python/current/tutorial.html#
```

```
from pymongo import MongoClient
```

```
mongo_uri = "mongodb://admin:superadmin1@ds253840.mlab.com:53840/c4e18-lab"
```

```
#1 connect database
```

```
client = MongoClient(mongo_uri)
```

```
# 2 get database
```

```
db = client.get_default_database()
```

```
# 3. Create collection
```

```
games = db["games"]
```



```
4. Create document
new_game = {
    "title": "PES",
    "description": "Pro evolution soccer"
}
```

```
5. Insert document
games.insert_one(new_game)
```

```
all_game = games.find()
for game in all_game:
    print(game)
```

## VII. Basic HTML

trong html sử dụng cặp thẻ (tag)  
bấm shift 1 để tạo phần below

<opening tag> CONTENT </closing tag>

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>HTML Intro</title>
```

```
</head>
```

```
<body>
```

Hello World

opening tag, closing tag

h1: heading (từ 1- 6)

```
<h1>Hello C4e18</h1>
```

```
<h3>Heading level 3</h3>
```

Thẻ p: Thẻ paragraph

```
<p> the paragraph</p>
```

```
<p>Lorem </p>
```

href: đường dẫn link, là 1 attribute, a: anchor  
attribute phải ở trong opening tab

```
<a href="http://dantri.com.vn/" target = "_blank">
```

Click vào em đi

```
</a>
```

```
<!-- division -->
```

thẻ b: bold  
thẻ div: chia tách

```
<div>
thẻ b: children of div, div: parent of <b></b>. Cấu trúc: Parent-Children-
Sibling
<b>Hello</b>
        <b>C4E18</b>
</div>

</body>
</html>
```

## VIII. Basic PyExcel

*# pip install pyeexcel*

```
import pyexcel
```

```
dictionary_1 = [
```

```
{
    "Name": 'Adam',
    "Age": 28
},
{
    "Name": 'Beatrice',
    "Age": 29
},
{
    "Name": 'Ceri',
    "Age": 30
},
{
    "Name": 'Dean',
    "Age": 26
}
]
```

```
pyexcel.save_as(records = dictionary_1, dest_file_name = "dictionary.xlsx")
```

## IX\_Basic Youtube downloading in Python

Note: If you have **not** installed youtube-dl yet, do it before trying this tutorial

```
pip install youtube-dl
```

```
from youtube_dl import YoutubeDL
```

Sample 1: Download a single youtube video

```
d1 = YoutubeDL()
```

```
d1.download(['https://www.youtube.com/watch?v=WHK5p7JL7g4'])
```

Remember to put your video **in** a **list**, eventhough one video **is** downloaded

Sample 2: Download multiple youtube videos

```
d1 = YoutubeDL()
```

Put **list** of song urls **in** download function to download them, one by one

```
d1.download(['https://www.youtube.com/watch?v=wNVIn-QS4DE',  
'https://www.youtube.com/watch?v=JZjRrg2rpic'])
```

Sample 3: Download audio

```
options = {
```

```
    'format': 'bestaudio/audio'
```

```
}
```

Tell the downloader to download only the best quality of audio

```
d1 = YoutubeDL(options)
```

```
d1.download(['https://www.youtube.com/watch?v=c3jHlYsnEe0'])
```

Sample 4: Search **and** then download the first video

```
options = {
```

```
    'default_search': 'ytsearch',
```

```
    'max_downloads': 1
```

```
}
```

tell downloader to search instead of directly downloading

Tell downloader to download only the first **entry** (video)

```
d1 = YoutubeDL(options)
```

```
d1.download(['con điền TAMKA PKL'])
```

Sample 5: Search **and** then download the first audio

```
options = {
```

```
    'default_search': 'ytsearch',
```

```
    'max_downloads': 1,
```

```
    'format': 'bestaudio/audio'
```

```
}
```

tell downloader to search instead of directly downloading

Tell downloader to download only the first **entry** (audio)

```
dl = YoutubeDL(options)
dl.download(['Nhớ mưa Sài Gòn lam trường'])
```

## X\_ Basic web scrapping from financial website (Cafef) - To review again

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import pyexcel

url = 'http://s.cafef.vn/bao-cao-tai-chinh/VNM/IncSta/2017/3/0/0/ket-qua-hoat-dong-kinh-doanh-cong-ty-co-phan-sua-viet-nam.chn'
html_content = urlopen('http://s.cafef.vn/bao-cao-tai-chinh/VNM/IncSta/2017/3/0/0/ket-qua-hoat-dong-kinh-doanh-cong-ty-co-phan-sua-viet-nam.chn').read().decode('utf-8')

soup = BeautifulSoup(html_content, 'html.parser')
table = soup.find('table', id='tableContent')

td_list = table.find_all("td", "b_r_c")

record = []
for i in range (0, len(td_list)-4, 6):
    data={}
    data['Content'] = td_list[i].string
    data['Quý 2-2017'] = td_list[i+3].string
    data['Quý 3-2017'] = td_list[i+4].string
    data['Quý 4-2017'] = td_list[i+1].string
    data['Quý 1-2018'] = td_list[i+2].string
    record.append(data)

pyexcel.save_as(records=record, dest_file_name="vinamilk.xlsx")
```

## XI\_Functions in Python

Làm chương trình 3 tầng: Input - Process - Output

### Define function:

```
def calculation(x, y, z):
    (x, y, z: Function parameters/ arguments)
    Variable in scope: Chỉ trong scope mới biết. (Scope: Chức năng hoạt động của function)
```

### Import function:

Cách 1:

```
from file_name import function
```

Cách 2:

```
Import file_name
```

### Use function:

Cách 1:

```
variable_x = file_name.function(x,y,z)
```

Cách 2:

```
variable_x = function(x,y,z)
```

### Return function

Return: Return [a, b, c]

### Call function:

```
calculation(x ,y , z)
```

Naming convention:

- function: Verb (lower case)
- Variable: Noun (lower case)
- Status: Adjective

## XII\_Web part I: Tạo server bằng local host

Viết server: fapp + tab

Viết html: ! + tab

Trong Jinja2: Gọi ra dictionary

```
list_1 = [{dictionary 1}, {dictionary 2}, ...]
```

```
list_1[key] = value
```

Install flask-snippets

Each function & route must have different names and not repeated

```
from jinja2 import Template
from flask import Flask
from flask import render_template
```

```
app = Flask(__name__)
```

Example 1: tạo trang chủ

```
@app.route('/')
def index():
    return "Hello World"
```

Example 2: Tạo subpage của trang chủ

```
@app.route("/hello")
def say_hello():
    return "Hello C4e18"
```

Example 3: Tạo subpage dùng parameters (1/2)

```
@app.route("/sayhitoneighbor/<name>/<age>")
def hello_neighnot(name, age):
    return "Hi {0} you are {1} years old".format(name, age)
```

example 4: Tạo subpage dùng parameters (2/2)

```
@app.route("/yolo/<meaning>")
def yolo(meaning):
    return "Its mean is" + meaning
```

Note: Nếu thêm parameters vào, request gửi về có bao nhiêu parameters thì function phải có bằng đấy

Example 5: Calculate SUM

```
@app.route("/sum/<int:x>/<int:y>")
def sum_calculate(x,y):
    return str(x + y)
```

Example 6: Sử dụng html

```
@app.route("/html")
def html_ex():
    return "<h1>hello Nam</h1>"
```

Example 7: Sử dụng html và lưu biến. Tao folder ten "templates"

```
@app.route("/index")
def html_ex2():
    post_title = "Thơ con ếch" # bien ben html: post_title
    post_content = "Tiền không mang lại hạnh phúc cho những người không có chúng"
    post_author = "Dao Hoang Nam"
    return render_template("index.html", post_title = post_title, post_content
= post_content, post_author = post_author)
```

Note: Biến trong file HTML (folder templates): {{Biến\_server\_gửi\_sang HTML}}

Định dạng: dictionary.

Viết tắt: list.key (list là tập con của lists)

Example 8: Sử dụng html và lưu biến. Server gửi nội dung sang html

```
@app.route("/index2")
def html_ex3():
    posts = [
        {
            "title": "Thơ con dế", # biến ben html: post_title
            "content": "Hôm nay trăng lên cao tí",
            "author": "thedarkknight513",
            "gender": 1
        },

        {"title": "Thơ con mèo", # biến ben html: post_title
         "content": "Nếu tiền không làm bạn hạnh phúc thì hãy đưa nó cho tôi",
         "author": "Nathan Drake",
         "gender": 0
        },

        {"title": "Thơ con dế", # biến ben html: post_title
         "content": "Đề thi vừa sức với các bạn và quá sức với em",
         "author": "python",
         "gender": 1
        }
    ]

    first_post = posts[1]
    C1:post là biến gửi sang html. Python thấy render template thì tự vào
    folder templates tìm
    return render_template("index.html", post = first_post)
    C2:
    return render_template("index.html", posts = posts)
    Bên TAB HTML:
    <body>
    {% for p in posts %}
    <h1>hello from the other side</h1>
    <h3>
        {{ p.title }}
    </h3>
    <p>
        {{ p.content}}
    </p>
    <i>
        by:
        {% if p.gender == 1 %}
        Mr.
        {% elif p.gender ==0 %}
        Ms.
        {% endif %}
        {{ p.author}}
```

```
{% endfor %}  
</body>
```

Luôn để run app ở cuối cùng. Khi mà file được chạy trực tiếp thì sẽ chạy những gì ở trong này. Mỗi khi khởi động server, nếu server có thay đổi gì thì sẽ tự động cập nhật ngay.

Host: Địa chỉ server. Port: Đường vào server. Domain = Host + Port

```
if __name__ == '__main__':  
    app.run(debug=True)
```