1. **General knowledge**

1.  dir: List các file trong folder

2.  Dấu “=”: Assign giá trị

“==”: Bolean 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")

(ban viet tat ngan gon (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 bát súp (trong trường hợp này là thẻ ul)

(tìm thẻ ul, class = "ul1 ulnew")

sử dụng: find/find\_all: Ten the - dac diem (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

lsit\_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 vao em di

</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

dl = YoutubeDL()

dl.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

dl = YoutubeDL()

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

dl.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

dl = YoutubeDL(options)

dl.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)

dl = YoutubeDL(options)

dl.download(['con điên 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 fuction:**

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, redirect

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", *# bien ben html: post\_title*

"content": "Hôm nay trăng lên cao tít",

"author": "thedarkknight513",

"gender": 1

},

{"title": "Thơ con mèo", *# bien 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", *# bien 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)

Example 9: Redirect to website (remember to import redirect)

@app.route('/school')

def redirect\_to\_url():

return redirect("http://techkids.vn")

**XIII. HTML**

W3school.com

Thẻ div: Thẻ block, chiếm nguyên 1 dòng

Thẻ span: Không chiếm 1 khối, chỉ chiếm nội dung nó đến thôi (thẻ in-line)

Thẻ br: Xuống dòng (break)

Thẻ hr: Gạch chia trang

Thẻ chia trang khác: section, header, footer, aside, article