**1. Gearing up with Python**

- List có giá trị vùng nhớ liên tiếp

- Python làm việc trực tiếp trên vùng nhớ máy tính, tên biến đại diện cho địa chỉ vùng nhớ

- pip dùng cài các library từ bên ngoài, syntax: pip install ….

> cài jupyter: ấn pip install jupyterlab

> mở project: cd vào folder chứa project, ghi jupyter lab

> hoặc dùng vscode

- Ấn A tạo 1 line above B 1 line below, ấn dd delete, ấn M tạo Markdown, **#** để phóng bự title trong markdown, shift enter để chạy

**2. Basic Data Types**

- Defining a variable: a = 5, variable name is normally lowercase

- Numbers: float, int, complex ( i ~ j phần ảo)

A screenshot of a computer

Description automatically generated

- Strings: nối string bằng dấu +, string + number bằng cách (strings, num) sẽ tự động có “ “ sau 2 thành phần.

- Booleans: True and False

**3. Data Structures**

- **Lists []:** có thể chứa multiple data types, có chỉ số, có thứ tự, ví dụ: [1,2] == [1,2] true nhưng [1,2] == [2,1] là false.

+ khởi tạo bằng list() hoặc []

+ slicing giống cới string [start:end:step]

+ insert(pos,value), pop() xóa phần tử cuối

+ dùng copy() để tạo ra 2 list khác nhau, không tham chiếu cùng 1 vùng nhớ

**- Sets {}:** tất cả các phần tử trong set là unique, không chỉ số, không thứ tự, có thể thêm phần tử ví dụ: {1,2} == {1,2} true và {1,2} == {2,1} là true.

+ khởi tạo bằng set() hoặc {}

+discard(value) xóa value cụ thể trong set

- Tuples (): không thể sửa giá trị, không thể thêm phần tử, có thứ tự, có chỉ số

+ dùng cho 1 dài tham số giá trị trả về. ví dụ return 1,2,3

- Dictionaries {key:value} : dict[key] = value

+ Vòng lặp:

for key in dict: hoặc for key in dict.keys():

for key,val in dict.items():

for val in dict.values():

**4. Arithmetic Operators**

- Arithmetic Operators: +, -, \*, /, //, %,….

- Arithmetic Operators with string: string + string đễ nối chuổi, string \* int để nhân chuỗi lên int lần.

**5. Comparasion, logical, membership operators**

- Comparision operators: >, <, >=, <=, ==, !=,…

- Logical operators: and, not, or

- Membership operators: in, not in

**6. Control Flow**

***If/else statements:***

*Syntax:*

If condition:

#statement

*Or*

if condition:

#statement

else:

#statement

*Or*

if condition:

#statement

elif condition:

#statement

else:

#statement

***For loops:***

*Syntax:*

for item in (container or range(:

#statement

***While loops:***

*Syntax:*

while condition:

#statement

**7. Function**

def func(parameters):

return value

**6. Classes**

***Syntax:***

class class\_name:

#luôn được gọi khi khởi tạo đối tượng (constructor), self is instance

def \_\_init\_\_(self){

#statement

- ‘.’ để truy cập thuộc tính (self.name ….)

- public (name), protected (\_name), private (\_\_name)

- tất cả method trong class đều có tham số self

Bài tập: Tính n!

A screenshot of a computer code

Description automatically generated

**8. Basic Data Types**

- Int and float: round() để làm tròn, int(float\_num) lấy phần nguyên, getContext().prec=4 set phần thập phân đằng sau chỉ lấy 4 số, int(string, <hexa,bin,…>) convert sang số nguyên

- Booleans: có 2 giá trị true và false, bool(something\_empty) trả về false ngược lại true

- Strings: có chỉ số, slicing [start:end:step], len() xem độ dài, formatting (f”string {value}” hoặc string.format(parameter)

**9. Bytes**

- bytes(tham\_so\_str, ‘<dinhdang utf-9..>), dùng decode(‘dinhdang’) giải mã ra

enc = bytes('😃','utf-8')

print(enc.decode('utf-8'))

- bytes(4) do creates an empty bytes object 4 bytes long.

Bài tập: chuyển hexa sang decimal

A screenshot of a computer code

Description automatically generated

**10. List Comprehensions**

[sth\_u\_do for item in list] 🡪 return a list

- List comprehension with filter: [sth\_u\_do for item in list condition] 🡪 [item for item in list if item > 10]

- List comprehension with functions: [your\_func for item in container condition]

- Nested list comprehension: [func for item in val for val in string]

How would you write a nested list comprehension to apply a function someFunc to all comments in a list of blog objects?

* [for blog in blogs [someFunc(comment) for comment in blog.comments]]
* [[someFunc(comment) for comment in blog.comments] for blog in blogs]
* [[blog for blog in blogs] for someFunc(comment) in blog.comments]
* [for someFunc(comment) in comments [blog.comments for blog in blogs ]]

**11. Dictionary comprehension**

- Giống list

Bài tập:

A screenshot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generated