

Python 3 Tham Khảo Nhanh

Cơ bản đến pandas, requests, csv, json

Cơ Bản

Biến

```
name = "Alice" # str
age = 20 # int
gpa = 3.85 # float
active = True # bool
```

Kiểu Dữ Liệu

```
str Chuỗi: "hello"
int Số nguyên: 42
float Số thực: 3.14
bool True / False
list Có thứ tự, thay đổi được: [1, 2, 3]
tuple Có thứ tự, bất biến: (1, 2)
dict Key-value: {"a": 1}
set Phần tử duy nhất: {1, 2, 3}
```

Toán Tử Số Học

```
+ - * Cộng, trừ, nhân
/ Chia (float): 7/2 → 3.5
// Chia lấy phần nguyên: 7//2 → 3
% Chia lấy dư: 7%2 → 1
** Lũy thừa: 2**3 → 8
```

Chuyển Đổi Kiểu

```
int("42") # 42
float("3.14") # 3.14
str(100) # "100"
list("abc") # ['a', 'b', 'c']
```

Nhập Liệu Người Dùng

```
name = input("Your name? ")
age = int(input("Age? "))
```

Chuỗi

Tạo Chuỗi

```
s1 = 'single quotes'
s2 = "double quotes"
s3 = """triple quotes
for multiline"""
```

f-String (Python 3.6+)

```
name = "Alice"
f"Hello, {name}!" # Hello, Alice!
f"{2 + 3}" # 5
f"{3.14159:.2f}" # 3.14
f"{1000:,"} # 1,000
```

Cắt Chuỗi

```
s = "Python"
# Index: 0 1 2 3 4 5
s[0] # 'P'
s[-1] # 'n'
s[2:5] # 'tho'
s[:2] # 'Py'
s[2:] # 'thon'
s[::-1] # 'nohtyP' (reverse)
```

Phương Thức Chuỗi

```
len(s) Độ dài chuỗi
s.upper() CHỮ HOA
s.lower() chữ thường
s.strip() Xóa khoảng trắng đầu/cuối
s.split(",") Tách thành list
", ".join(lst) Nối list thành chuỗi
s.replace(a, b) Thay thế a bằng b
s.find("x") Chỉ số đầu tiên (-1 nếu không có)
s.startswith(x) Kiểm tra tiền tố → bool
s.endswith(x) Kiểm tra hậu tố → bool
s.count(x) Đếm số lần xuất hiện
"x" in s Kiểm tra chứa → bool
```

List

Tạo & Truy Cập

```
fruits = ["apple", "banana", "cherry"]
fruits[0] # "apple"
fruits[-1] # "cherry"
fruits[1:3] # ["banana", "cherry"]
```

List Comprehension

```
squares = [x**2 for x in range(5)]
# [0, 1, 4, 9, 16]
evens = [x for x in range(10) if x%2==0]
# [0, 2, 4, 6, 8]
```

Phương Thức List

```
lst.append(x) Thêm vào cuối
lst.extend(lst2) Thêm tất cả phần tử từ lst2
lst.insert(i, x) Chèn vào chỉ số i
lst.pop() Xóa & trả về phần tử cuối
lst.pop(i) Xóa & trả về phần tử tại i
lst.remove(x) Xóa x đầu tiên
del lst[i] Xóa theo chỉ số
lst.sort() Sắp xếp tại chỗ
sorted(lst) Trả về bản sao đã sắp xếp
lst.reverse() Đảo ngược tại chỗ
len(lst) Số phần tử
x in lst Kiểm tra thành viên
lst.index(x) Chỉ số đầu tiên của x
lst.count(x) Số lần xuất hiện của x
```

Tuple & Set

Tuple (Bất Biến)

```
point = (3, 4)
x, y = point # unpacking
point[0] # 3 (read-only)
```

Set (Phần Tử Duy Nhất)

```
s = {1, 2, 3}
s.add(4); s.remove(1)
a & b # intersection
a | b # union
a - b # difference
```

Dictionary

Tạo & Truy Cập

```
student = {"name": "Alice", "age": 20}
student["name"] # "Alice"
student.get("gpa", 0) # 0 (default)
student["gpa"] = 3.85 # add/update
```

Dict Comprehension

```
sq = {x: x**2 for x in range(5)}
# {0:0, 1:1, 2:4, 3:9, 4:16}
```

Duyệt Qua

```
for k, v in student.items():
    print(f"{k}: {v}")
```

Phương Thức Dict

```
d.keys() Tất cả key
d.values() Tất cả value
d.items() Tất cả cặp (key, value)
d.get(k, default) Lấy giá trị với fallback
d.update(d2) Merge d2 vào d
d.pop(k) Xóa & trả về giá trị
del d[k] Xóa key
"key" in d Key tồn tại? → bool
len(d) Số phần tử
```

Làm Điều Khiển

if / elif / else

```
if score >= 90:
    grade = "A"
elif score >= 80:
    grade = "B"
else:
    grade = "C"
```

Toán Tử Ba Góc

```
status = "pass" if score >= 60 else "fail"
```

Vòng Lặp

Vòng Lặp for

```
for fruit in ["apple", "banana"]:
    print(fruit)
```

range()

```
range(5) # 0, 1, 2, 3, 4
range(2, 5) # 2, 3, 4
range(0, 10, 2) # 0, 2, 4, 6, 8
```

Vòng Lặp while

```
while count < 10:
    count += 1
```

enumerate() & zip()

```
for i, val in enumerate(["a", "b"]):
    print(i, val) # 0 a, 1 b
```

```
for a, b in zip([1, 2], ["x", "y"]):
    print(a, b) # 1 x, 2 y
```

break & continue

```
for x in range(10):
    if x == 5: break # stop loop
    if x % 2 == 0: continue # skip
```

Python 3 Tham Khảo Nhanh

Hàm

Định Nghĩa & Gọi Hàm

```
def greet(name, greeting="Hi"):
    return f"{greeting}, {name}!"

greet("Alice") # "Hi, Alice!"
greet("Bob", "Hello") # "Hello, Bob!"
```

Nhiều Giá Trị Trả Về

```
def min_max(lst):
    return min(lst), max(lst)
lo, hi = min_max([3, 1, 4, 1, 5])
```

*args & **kwargs

```
def total(*args): # args is a tuple
    return sum(args)
total(1, 2, 3) # 6

def info(**kwargs): # kwargs is a dict
    print(kwargs)
```

Lambda

```
square = lambda x: x**2
square(5) # 25
sorted(lst, key=lambda x: x["age"])
```

Class

```
class Dog:
    def __init__(self, name, breed):
        self.name = name
        self.breed = breed

    def bark(self):
        return f"{self.name} says Woof!"

dog = Dog("Rex", "Lab")
dog.bark() # "Rex says Woof!"
```

Kế Thừa

```
class Puppy(Dog):
    def __init__(self, name, breed, toy):
        super().__init__(name, breed)
        self.toy = toy
```

Xử Lý Lỗi

```
try:
    result = 10 / 0
except ZeroDivisionError as e:
    print(f"Error: {e}")
finally:
    print("Always runs")
```

File I/O

Đọc File

```
with open("data.txt") as f:
    content = f.read() # full text

with open("data.txt") as f:
    for line in f: # line by line
        print(line.strip())
```

Ghi File

```
with open("out.txt", "w") as f:
    f.write("Hello\n")
```

"r" = read "w" = write (overwrite) "a" = append

refmint.com

CSV

```
import csv

with open("data.csv") as f:
    reader = csv.DictReader(f)
    for row in reader:
        print(row["name"])

with open("out.csv", "w", newline="") as f:
    writer = csv.writer(f)
    writer.writerow(["name", "age"])
```

JSON

```
import json

data = json.loads('{"name": "Alice"}') # parse
text = json.dumps(data) # serialize

with open("data.json") as f:
    data = json.load(f) # read file
with open("out.json", "w") as f:
    json.dump(data, f, indent=2) # write file
```

HTTP Request

```
import requests

# GET
r = requests.get("https://api.example.com/data")
r.status_code # 200
data = r.json() # parse JSON

# POST
r = requests.post(url, json={"key": "val"})
```

Pandas Cơ Bản

```
import pandas as pd
df = pd.read_csv("data.csv")
df.head() # first 5 rows
df.shape # (rows, cols)
df["name"] # single column
df[df["age"] > 20] # filter rows
```

Hàm Dựng Sẵn Hữu Ích

print()	In ra console
len()	Độ dài / số lượng
type()	Kiểu của đối tượng
range()	Dãy số
enumerate()	Cặp chỉ số + giá trị
zip()	Ghép phần tử từ các iterable
sorted()	Trả về bản sao đã sắp xếp
sum() min() max()	Hàm tổng hợp

Module

```
import math
from math import sqrt, pi
import pandas as pd # alias
```