



## CSC17106 – XỬ LÝ PHÂN TÍCH DỮ LIỆU TRỰC TUYẾN

### HƯỚNG DẪN THỰC HÀNH

#### PYTHON CƠ BẢN

#### I. Thông tin chung

Mã số:	HD01
Thời lượng dự kiến:	3 tiếng
Deadline nộp bài:	-
Hình thức:	-
Hình thức nộp bài:	-
GV phụ trách:	Phạm Minh Tú
Thông tin liên lạc với GV:	pmtu@fit.hcmus.edu.vn

#### II. Chuẩn đầu ra cần đạt

Bài hướng dẫn này nhằm mục tiêu đạt giúp sinh viên được các mục tiêu sau:

1. Cài đặt python và môi trường phát triển
2. Sử dụng python mức cơ bản
3. Sử dụng python để thao tác với tập tin

#### III. Mô tả

##### 1. Cài đặt Python trên Window và môi trường phát triển

###### Bước 1: Tải xuống trình cài đặt Python:

Truy cập trang web chính thức của Python tại <https://www.python.org/downloads/> và tải xuống phiên bản Python mới nhất cho Windows. Hiện tại, Python 3.x là phiên bản mới nhất, nên chọn phiên bản này.

###### Bước 2: Chạy trình cài đặt Python:

Sau khi tải xong, mở tệp tải về (thường có đuôi là .exe) và chạy nó. Trong trình cài đặt Python, nhớ chọn các tùy chọn sau:

Chọn "Add Python x.x to PATH" (x.x là phiên bản Python bạn đang cài đặt). Điều này sẽ cho phép bạn sử dụng Python từ Command Prompt dễ dàng hơn.

Chọn "Customize installation" để tùy chỉnh cài đặt theo nhu cầu của bạn. Nếu bạn không có yêu cầu cụ thể, bạn có thể giữ nguyên các tùy chọn mặc định.

###### Bước 3: Hoàn tất cài đặt:

Nhấn nút "Install Now" để bắt đầu quá trình cài đặt Python. Quá trình này sẽ mất một thời gian ngắn để hoàn thành.

#### Bước 4: Kiểm tra cài đặt:

Sau khi cài đặt hoàn tất, bạn có thể kiểm tra xem Python đã được cài đặt thành công bằng cách mở Command Prompt và gõ lệnh sau:

CSS

Copy code

```
python --version
```

Nếu bạn thấy phiên bản Python hiện ra, điều này có nghĩa là bạn đã cài đặt Python thành công trên Windows.

#### Bước 5 (tùy chọn): Cài đặt trình quản lý gói (Pip):

Python thường đi kèm với trình quản lý gói Pip. Bạn có thể kiểm tra phiên bản Pip của bạn bằng cách chạy lệnh sau trong Command Prompt:

CSS

Copy code

```
pip --version
```

Nếu Pip đã được cài đặt, bạn có thể bắt đầu cài đặt các thư viện và ứng dụng Python bằng cách sử dụng Pip.

Bây giờ bạn đã cài đặt Python thành công trên Windows và có thể bắt đầu phát triển ứng dụng Python trên máy tính của bạn.

Môi trường phát triển Python (Python development environment) là môi trường mà bạn sử dụng để viết, kiểm thử, và phát triển các ứng dụng Python. Dưới đây là một số yếu tố quan trọng mà bạn cần xem xét khi tạo môi trường phát triển Python:

- **Trình soạn thảo hoặc IDE (Integrated Development Environment):**

Trình soạn thảo là một ứng dụng giúp bạn viết mã Python. Một số ví dụ phổ biến là **Visual Studio Code**, **Sublime Text**, và **Atom**.

IDE là một môi trường tích hợp có nhiều tính năng hữu ích như chỉ mục code, gợi ý, kiểm tra lỗi, và nhiều tính năng khác. Một số IDE phổ biến cho Python bao gồm **PyCharm**, **Visual Studio**, và **Thonny**.

- **Trình quản lý gói (Package Manager):**

Trình quản lý gói giúp bạn cài đặt, quản lý và cập nhật các thư viện và gói phụ thuộc trong dự án Python của bạn. **Pip** là trình quản lý gói tiêu chuẩn cho Python.

- **Công cụ kiểm tra mã (Code Version Control):**

Công cụ kiểm tra mã giúp bạn theo dõi sự thay đổi trong mã nguồn của bạn và làm việc cùng nhau trong dự án. Git là một công cụ kiểm tra mã phổ biến và **GitHub**, **GitLab** và **Bitbucket** là các dịch vụ lưu trữ mã nguồn phổ biến.

- **Thư viện và Frameworks:**

Python có nhiều thư viện và frameworks mạnh mẽ để giúp bạn phát triển các ứng dụng. Ví dụ: **Django** và **Flask** cho phát triển web, **NumPy** và **pandas** cho xử lý dữ liệu, **Matplotlib** và **Seaborn** cho trực quan hóa dữ liệu, và nhiều thư viện khác.

## 2. Sử dụng python mức cơ bản

- **In ra màn hình (Print)**

Hãy bắt đầu với một chương trình Python đơn giản in ra màn hình:

```
print("Xin chào, Python!")
```

- **Biến và Kiểu Dữ Liệu**

Python hỗ trợ nhiều kiểu dữ liệu, ví dụ:

```
# Kiểu số nguyên (integer)
```

```
age = 25
```

```
# Kiểu số thực (float)
```

```
height = 1.75
```

```
# Kiểu chuỗi (string)
```

```
name = "Alice"
```

```
# Kiểu Boolean (bool)
```

```
is_student = True
```

- **Các Phép Toán Cơ Bản**

Python cho phép bạn thực hiện các phép toán cơ bản như cộng, trừ, nhân, chia:

```
a = 10
```

```
b = 5
```

```
addition = a + b          # Cộng
```

```
subtraction = a - b       # Trừ
```

```
multiplication = a * b    # Nhân
```

```
division = a / b          # Chia
```

```
print(addition, subtraction, multiplication, division)
```

- **Câu Lệnh Điều Kiện (If Statement)**

Câu lệnh điều kiện cho phép bạn kiểm tra một điều kiện và thực hiện các hành động khác nhau dựa trên kết quả của điều kiện:

```
age = 18

if age >= 18:

    print("Bạn đã đủ tuổi để lái xe.")

else:

    print("Bạn chưa đủ tuổi để lái xe.")
```

- **Vòng Lặp (Loops)**

Python hỗ trợ hai loại vòng lặp chính: vòng lặp for và vòng lặp while:

Vòng lặp for dùng để lặp qua một chuỗi hoặc danh sách các phần tử:

```
fruits = ["táo", "chuối", "cam"]

for fruit in fruits:

    print(fruit)
```

Vòng lặp while lặp cho đến khi một điều kiện được thỏa mãn:

```
count = 0

while count < 5:

    print(count)

    count += 1
```

- **Hàm (Function)**

Hàm là một cách để bạn đóng gói một phần mã thành một khối thực thi có thể được gọi bất kỳ lúc nào:

```
def greet(name):

    print(f"Xin chào, {name}!")

greet("Alice")
```

- **Các Thư Viện Mở Rộng**

**Python** có một hệ thống thư viện phong phú cho nhiều mục đích khác nhau. Ví dụ, bạn có thể sử dụng thư viện **math** để thực hiện các phép toán số học phức tạp, và thư viện **random** để tạo số ngẫu nhiên.

**Thư viện numpy:** Numpy là một thư viện mạnh mẽ cho xử lý mảng và ma trận số học. Nó thường được sử dụng trong các ứng dụng liên quan đến khoa học dữ liệu và tính toán số học.

```
import numpy as np

# Tạo một mảng numpy
arr = np.array([1, 2, 3, 4, 5])

# Thực hiện các phép toán trên mảng
mean = np.mean(arr)
std_dev = np.std(arr)

print(mean, std_dev)
```

**Thư viện requests:** Thư viện requests được sử dụng để gửi và nhận các yêu cầu HTTP. Điều này cho phép bạn tương tác với các API web và truy xuất dữ liệu từ trang web.

```
import requests

# Gửi yêu cầu GET đến một trang web
response = requests.get("https://www.example.com")

# In nội dung trang web
print(response.text)
```

### Xử lý chuỗi:

#### Nối chuỗi

```
str1 = "Hello"
str2 = "World"
result = str1 + " " + str2
print(result) # Kết quả: "Hello World"
```

#### Chuyển chữ thường – hoa

```
text = "Hello, World"
uppercase_text = text.upper()
lowercase_text = text.lower()
print(uppercase_text) # Kết quả: "HELLO, WORLD"
print(lowercase_text) # Kết quả: "hello, world"
```

### Tách chuỗi

```
text = "apple,banana,cherry"

fruits = text.split(",")

print(fruits)  # Kết quả: ['apple', 'banana', 'cherry']
```

### Định dạng chuỗi

```
name = "Alice"

age = 30

message = f"Xin chào, tôi là {name} và tôi {age} tuổi."

print(message)  # Kết quả: "Xin chào, tôi là Alice và tôi 30 tuổi."
```

### Xử lý mảng:

#### Thêm phần tử vào danh sách

```
fruits = ["apple", "banana", "cherry"]

fruits.append("orange")

print(fruits)  # Kết quả: ['apple', 'banana', 'cherry', 'orange']
```

#### Truy cập phần tử vào danh sách

```
fruits = ["apple", "banana", "cherry"]

first_fruit = fruits[0]

print(first_fruit)  # Kết quả: "apple"
```

#### Lặp qua danh sách

```
fruits = ["apple", "banana", "cherry"]

for fruit in fruits:

    print(fruit)

# Kết quả:

# "apple"

# "banana"

# "cherry"
```

#### Sắp xếp danh sách

```
numbers = [5, 2, 8, 1, 6]

numbers.sort()

print(numbers) # Kết quả: [1, 2, 5, 6, 8]
```

Loại bỏ phần tử trong danh sách

```
fruits = ["apple", "banana", "cherry"]

fruits.remove("banana")

print(fruits) # Kết quả: ['apple', 'cherry']
```

### 3. Sử dụng python để thao tác với tập tin

Thao tác với File CSV bằng Pandas:

Để đọc và viết file CSV bằng pandas, bạn cần cài đặt thư viện pandas trước dùng **pip install pandas** (nếu chưa có).

Đọc File CSV:

```
import pandas as pd

# Đọc file CSV và lưu dữ liệu vào DataFrame
df = pd.read_csv('data.csv')

# In ra một số dòng đầu tiên của DataFrame
print(df.head())
```

Ghi File CSV:

```
import pandas as pd

# Tạo một DataFrame
data = {'Name': ['Alice', 'Bob', 'Charlie'],
        'Age': [25, 30, 35]}
df = pd.DataFrame(data)

# Viết DataFrame ra file CSV
df.to_csv('output.csv', index=False) # index=False để không viết thêm cột
```

Thao tác với File Excel bằng thư viện openpyxl:

Đọc File Excel:

```
import openpyxl

# Mở một file Excel
workbook = openpyxl.load_workbook('data.xlsx')

# Chọn một sheet
sheet = workbook.active # hoặc sử dụng sheet = workbook['Tên_Sheet']

# Đọc dữ liệu từ sheet
for row in sheet.iter_rows(values_only=True):
    print(row)
```

Ghi file Excel



```
import openpyxl

# Tạo một file Excel mới
workbook = openpyxl.Workbook()
sheet = workbook.active

# Thêm dữ liệu vào sheet
data = [['Name', 'Age'],
        ['Alice', 25],
        ['Bob', 30],
        ['Charlie', 35]]

for row_data in data:
    sheet.append(row_data)

# Lưu file Excel
workbook.save('output.xlsx')
```

#### IV. Tài liệu tham khảo

[Trang chính thức của Python](#)  
[W3Schools Python Tutorial](#)  
[Tài liệu chính thức của Pandas](#)  
[Tài liệu chính thức của NumPy](#)  
[Python.org Beginners Guide](#)  
[Real Python](#)  
[Geeks for Geeks Python Programming Language](#)  
[Stack Overflow Python Tag](#)

#### V. Bài tập

1. Tính tổng của các số lẻ từ 1 đến 100.
2. Viết một chương trình Python để in ra tất cả các số chia hết cho 7 nhưng không phải là bội số của 5 trong khoảng từ 2000 đến 3200 (bao gồm cả 2000 và 3200).
3. Viết một chương trình Python để đảo ngược một chuỗi nhập từ bàn phím. Ví dụ: "Hello World" sẽ trở thành "dlroW olleH".
4. Viết một chương trình Python để tính giai thừa của một số nguyên dương được nhập từ bàn phím. Giai thừa của n là  $n! = 1 * 2 * 3 * \dots * n$ .
5. Viết một chương trình Python để tạo một danh sách gồm các số nguyên từ 1 đến 50 và in ra các số chia hết cho 3 và 5 dưới dạng "FizzBuzz", các số chia hết cho 3 dưới dạng "Fizz", và các số chia hết cho 5 dưới dạng "Buzz".
6. Viết một chương trình Python để kiểm tra xem một chuỗi có phải là chuỗi đối xứng hay không. Ví dụ: "radar" và "level" là chuỗi đối xứng, trong khi "hello" không phải.
7. Viết một chương trình Python để xác định số lớn nhất trong một danh sách số nguyên nhập từ bàn phím.





8. Viết một chương trình Python để xóa các phần tử trùng lặp trong một danh sách và in ra danh sách sau khi đã loại bỏ các phần tử trùng lặp.
9. Viết một chương trình Python để kiểm tra xem một số có phải là số nguyên tố hay không. Số nguyên tố là số chỉ có hai ước số dương là 1 và chính nó.
10. Viết một chương trình Python để tính tổng của tất cả các số trong một danh sách hoặc tuple nhập từ bàn phím.
11. Đọc và Hiển Thị Dữ Liệu từ File CSV:
  - Yêu cầu người dùng nhập tên file CSV từ bàn phím.
  - Đọc dữ liệu từ file CSV và lưu vào một danh sách hoặc DataFrame (sử dụng thư viện pandas).
  - In ra dữ liệu từ file CSV lên màn hình.
12. Tính Tổng hoặc Trung Bình của Dãy Số từ File CSV:
  - Yêu cầu người dùng nhập tên file CSV và tên cột chứa dãy số từ bàn phím.
  - Đọc dữ liệu từ file CSV và lưu vào một danh sách hoặc DataFrame.
  - Tính tổng hoặc trung bình của dãy số và in ra kết quả.
13. Thêm Dữ Liệu Vào File CSV:
  - Yêu cầu người dùng nhập tên file CSV từ bàn phím và dữ liệu mới cần thêm (ví dụ: tên, điểm số).
  - Đọc dữ liệu hiện có từ file CSV và lưu vào một danh sách hoặc DataFrame.
  - Thêm dữ liệu mới vào danh sách hoặc DataFrame.
  - Ghi lại danh sách hoặc DataFrame mới vào file CSV.