

# Kỹ thuật lập trình với Python: XỬ LÝ DỮ LIỆU TẬP TIN

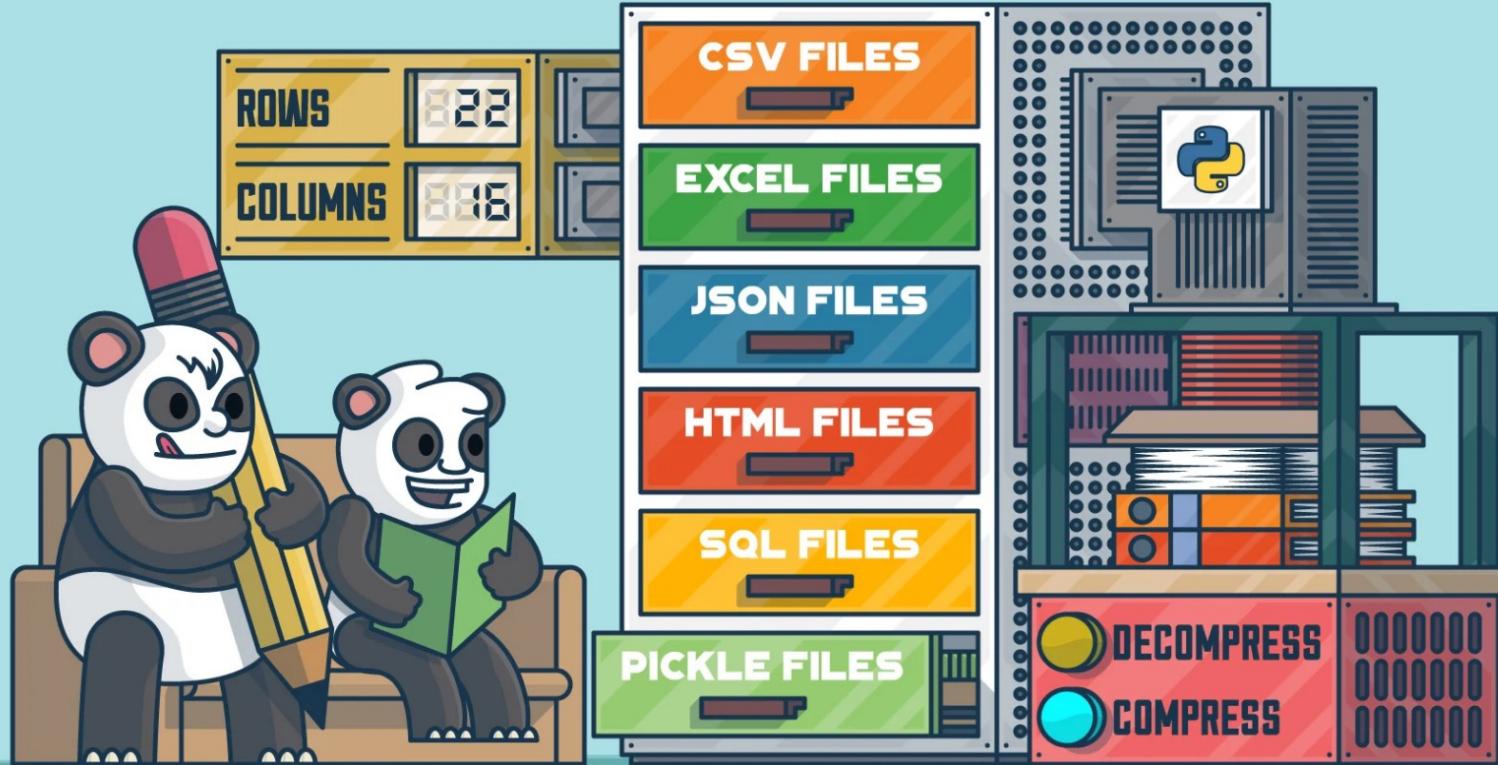
ThS. Nguyễn Quang Phúc  
[phucnq@uel.edu.vn](mailto:phucnq@uel.edu.vn)



# NỘI DUNG

1. Tập tin dữ liệu
2. Xử lý tập tin (file)

# 1. Tập tin dữ liệu



Real Python

# 1. Tập tin dữ liệu



Real Python

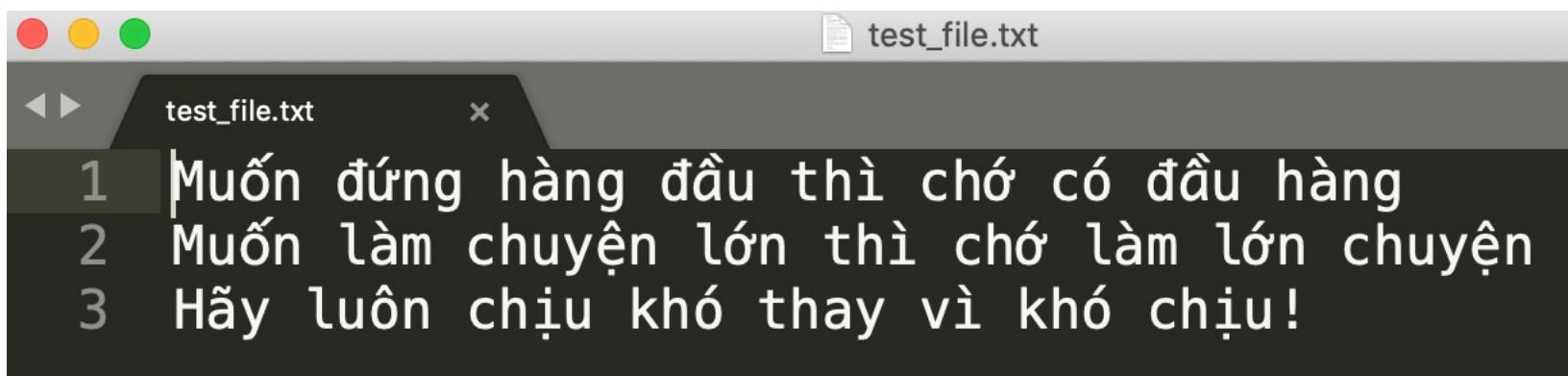
## 2. Xử lý tập tin (file)

### »» Text file (.txt)

#### Ghi dữ liệu

w (write): ghi đè dữ liệu  
a (append): ghi nối tiếp dữ liệu  
...

```
f = open('test_file.txt', 'w', encoding='utf-8')
f.write("Muốn đứng hàng đầu thì chờ có đầu hàng\n")
f.write("Muốn làm chuyện lớn thì chờ làm lớn chuyện\n")
f.write("Hãy luôn chịu khó thay vì khó chịu!")
f.close()
```



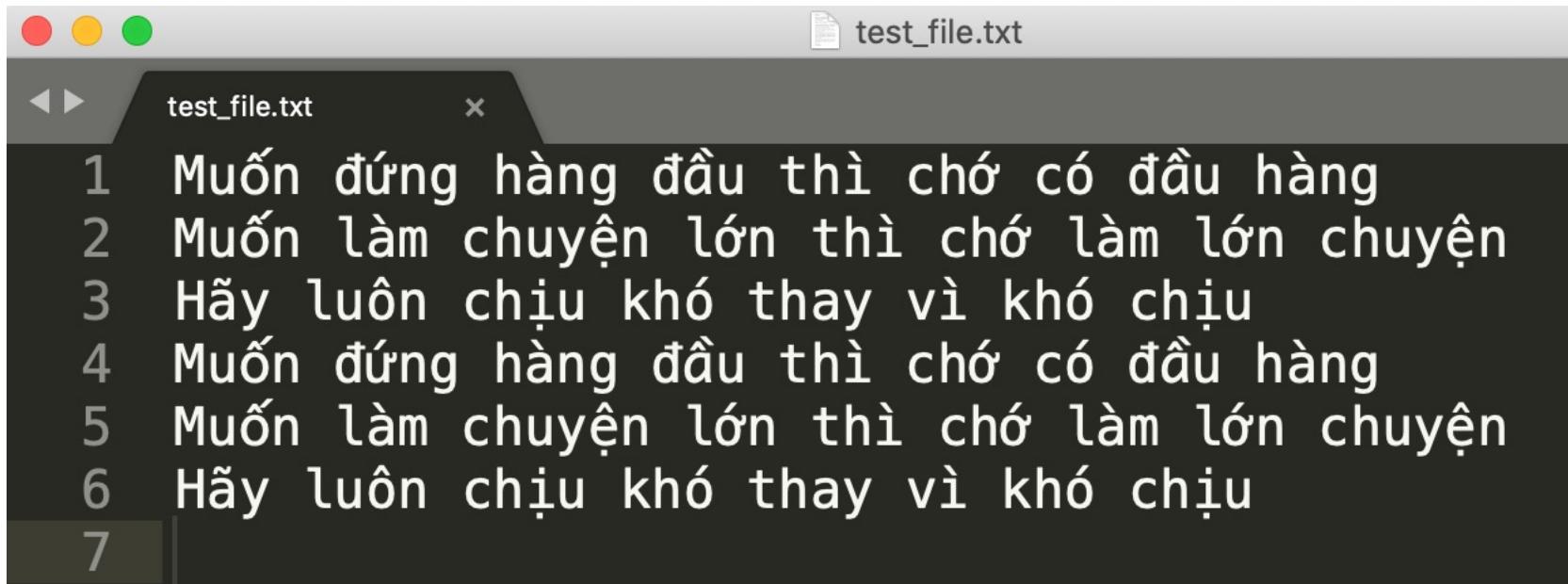
## 2. Xử lý tập tin (file)

### »» Text file (.txt)

#### Ghi dữ liệu

w (write): ghi đè dữ liệu  
a (append): ghi nối tiếp dữ liệu  
...

```
with open('test_file.txt', 'a', encoding="utf-8") as f:  
    f.write("Muốn đứng hàng đầu thì chờ có đầu hàng\n")  
    f.write("Muốn làm chuyện lớn thì chờ làm lớn chuyện\n")  
    f.write("Hãy luôn chịu khó thay vì khó chịu\n")
```



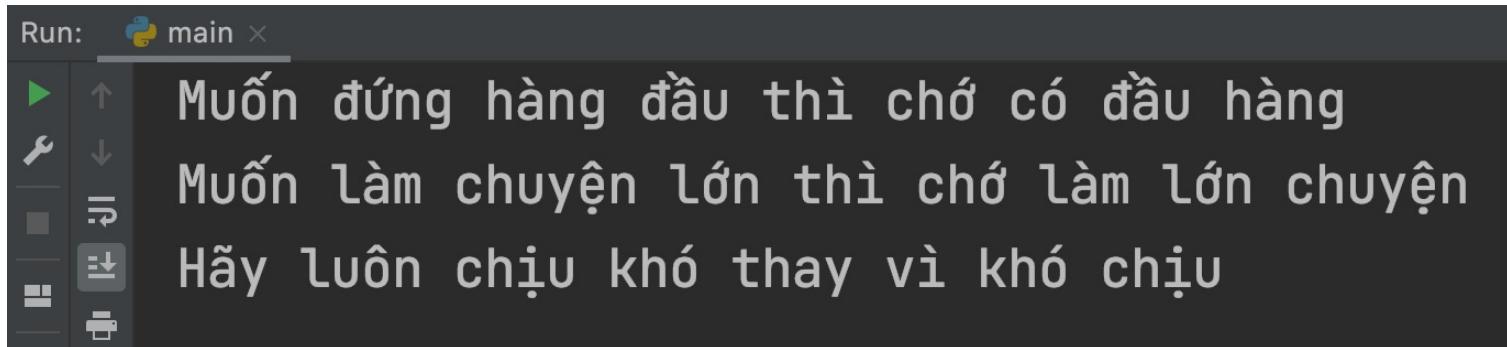
## 2. Xử lý tập tin (file)

### »»» Text file (.txt)

#### Đọc dữ liệu

```
f = open('test_file.txt', 'r', encoding="utf-8")
for line in f:
    print(line.strip())
f.close()
```

```
with open('test_file.txt', 'r', encoding="utf-8") as f:
    print(f.read())
```



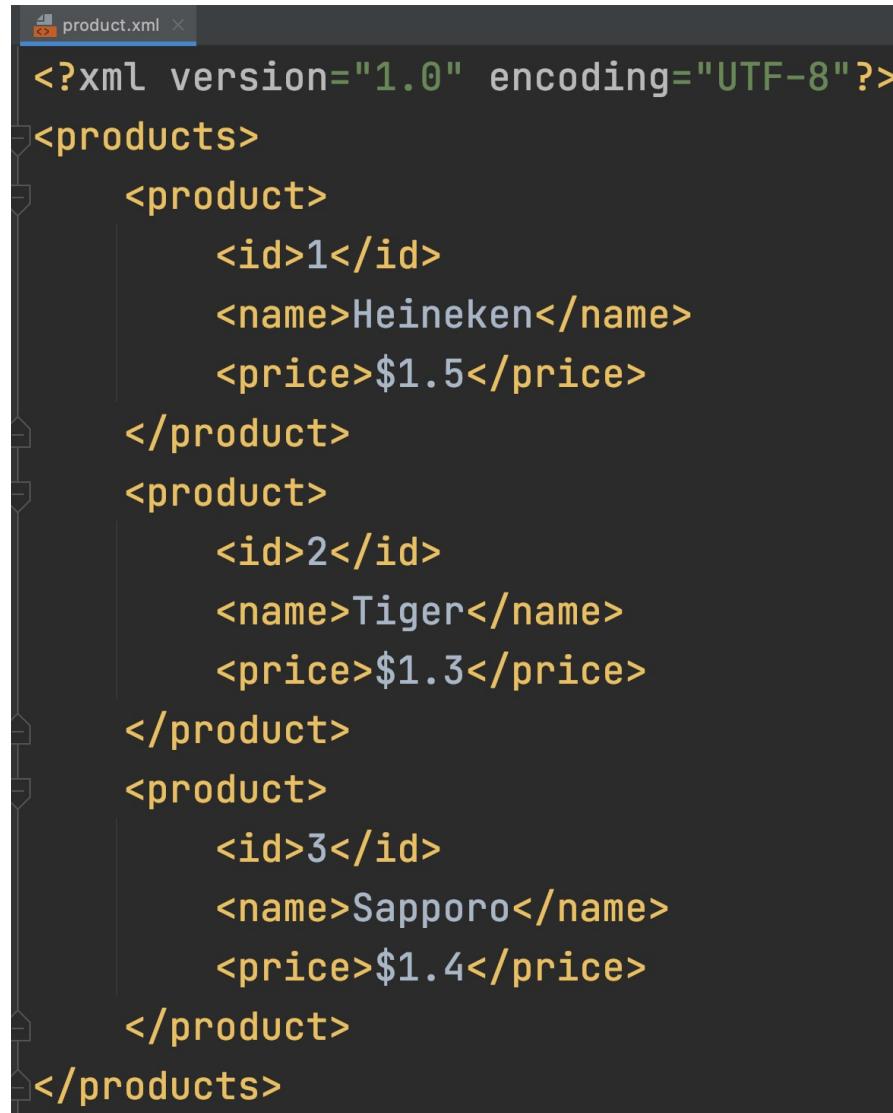
The screenshot shows a code editor interface with a dark theme. On the left, there is a toolbar with various icons for file operations like run, save, and copy. The main window displays a Python script named 'main'. The output pane shows the following text:

```
Muốn đứng hàng đầu thì chờ có đầu hàng
Muốn làm chuyện lớn thì chờ làm lớn chuyện
Hãy luôn chịu khó thay vì khó chịu
```

## 2. Xử lý tập tin (file)

### » XML file (.xml)

Vd: cấu trúc  
File XML



```
<?xml version="1.0" encoding="UTF-8"?>
<products>
    <product>
        <id>1</id>
        <name>Heineken</name>
        <price>$1.5</price>
    </product>
    <product>
        <id>2</id>
        <name>Tiger</name>
        <price>$1.3</price>
    </product>
    <product>
        <id>3</id>
        <name>Sapporo</name>
        <price>$1.4</price>
    </product>
</products>
```

## 2. Xử lý tập tin (file)

### » XML file (.xml)

```
from xml.dom.minidom import parse
```

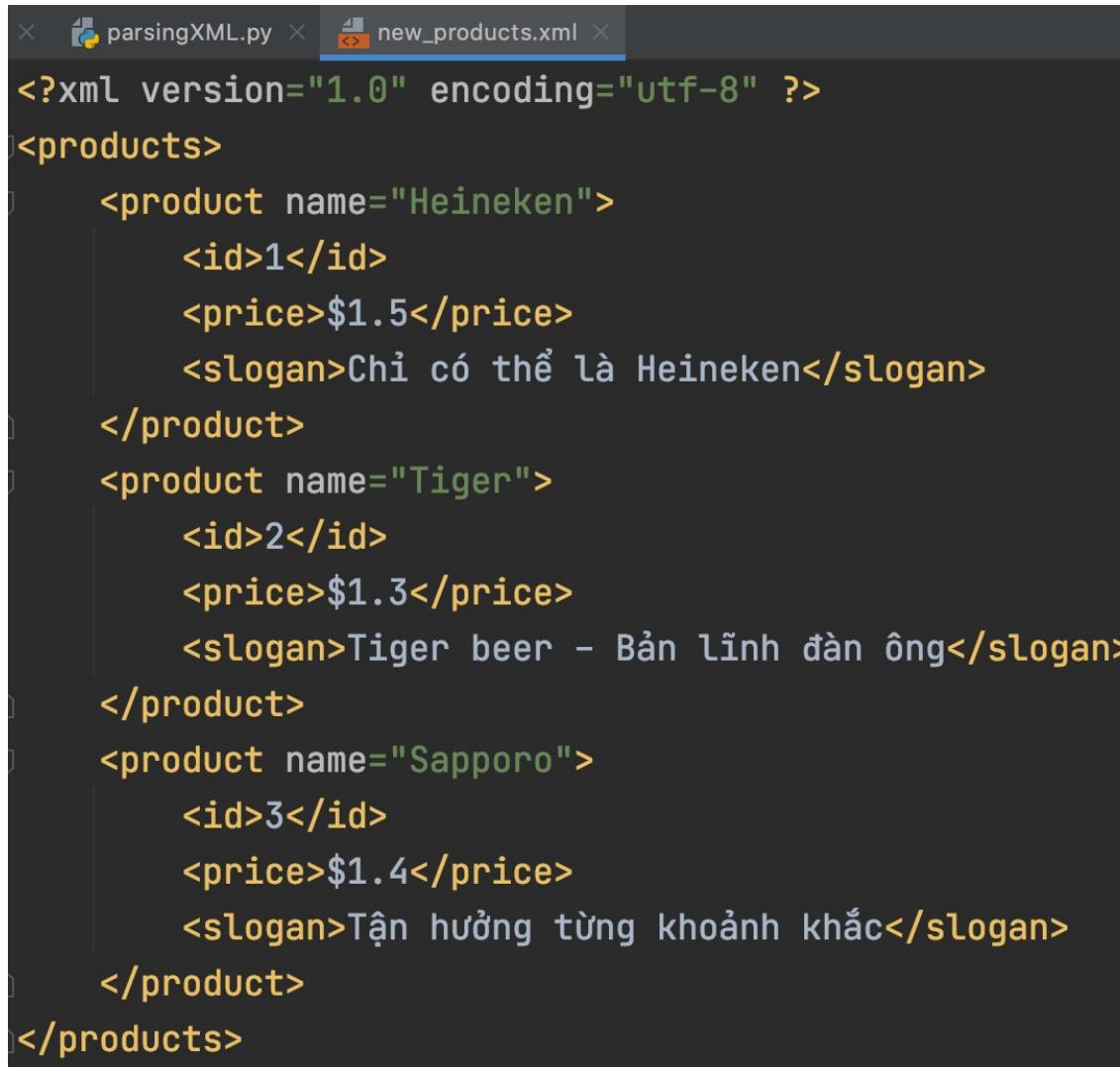
```
DOMTree = parse("product.xml")
elements = DOMTree.documentElement
```

```
products = elements.getElementsByTagName("product")
for p in products:
    pro_id = (p.getElementsByTagName("id")[0]).childNodes[0].data
    pro_name = (p.getElementsByTagName("name")[0]).childNodes[0].data
    pro_price = (p.getElementsByTagName("price")[0]).childNodes[0].data
    print(pro_id + " -> " + pro_name + " -> " + pro_price)
```

```
1 -> Heineken -> $1.5
2 -> Tiger -> $1.3
3 -> Sapporo -> $1.4
```

## 2. Xử lý tập tin (file)

### » XML file (.xml)



The screenshot shows a code editor window with two tabs: "parsingXML.py" and "new\_products.xml". The "new\_products.xml" tab is active, displaying the following XML code:

```
<?xml version="1.0" encoding="utf-8" ?>
<products>
    <product name="Heineken">
        <id>1</id>
        <price>$1.5</price>
        <slogan>Chỉ có thể là Heineken</slogan>
    </product>
    <product name="Tiger">
        <id>2</id>
        <price>$1.3</price>
        <slogan>Tiger beer - Bản lĩnh đàn ông</slogan>
    </product>
    <product name="Sapporo">
        <id>3</id>
        <price>$1.4</price>
        <slogan>Tận hưởng từng khoảnh khắc</slogan>
    </product>
</products>
```

## 2. Xử lý tập tin (file)

### » XML file (.xml)

```
import xml.etree.ElementTree as ET
tree = ET.parse("new_products.xml")
root = tree.getroot()
print("root_tag: ", root.tag)
print("root_attribute: ", root.attrib)
for child in root:
    print(child.tag, child.attrib)
print(root[1][2].text)
for p in root.iter('product'):
    print(p.attrib)
for p in root.findall('product'):
    p_id = p.find('id').text
    p_name = p.get('name')
    p_price = p.find('price').text
    p_slogan = p.find('slogan').text
    print(p_id + " - " + p_name + " - " + p_price + " - " + p_slogan + "")
```

```
root_tag: products
root_attribute: {}
product {'name': 'Heineken'}
product {'name': 'Tiger'}
product {'name': 'Sapporo'}
Tiger beer - Bán lĩnh đòn ông
{'name': 'Heineken'}
{'name': 'Tiger'}
{'name': 'Sapporo'}
1 - Heineken - $1.5 - 'Chỉ có thể là Heineken'
2 - Tiger - $1.3 - 'Tiger beer - Bán lĩnh đòn ông'
3 - Sapporo - $1.4 - 'Tận hưởng từng khoảnh khắc'
```

## 2. Xử lý tập tin (file)

### » XML file (.xml)

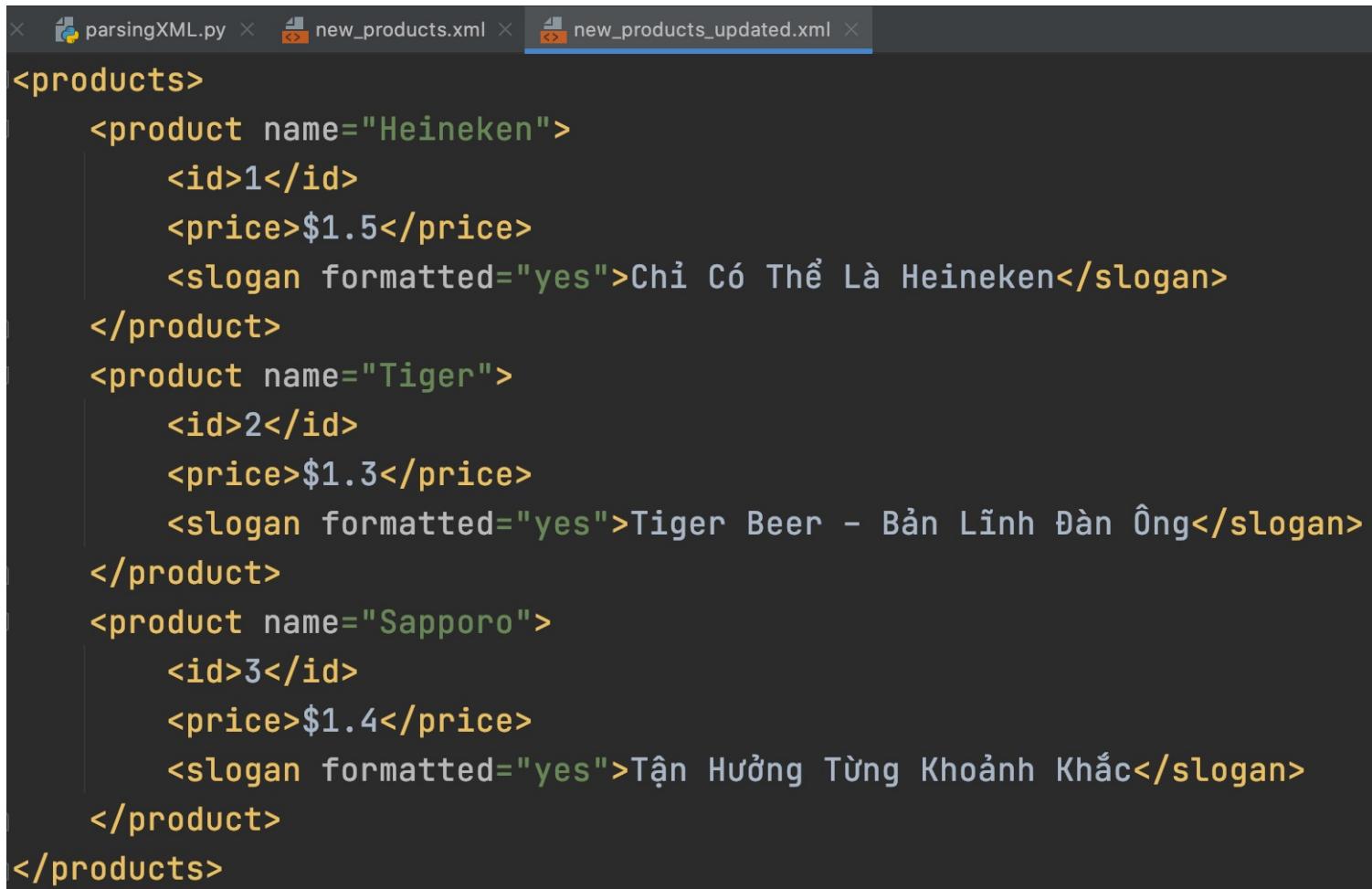
#### Modify XML file

```
import xml.etree.ElementTree as ET
tree = ET.parse("new_products.xml")
root = tree.getroot()
for slogan in root.iter('slogan'):
    formatted_slogan = slogan.text.title()
    slogan.text = formatted_slogan
    slogan.set('formatted', 'yes')
tree.write('new_products_updated.xml', encoding='utf-8')
```

## 2. Xử lý tập tin (file)

### » XML file (.xml)

#### Modify XML file



The screenshot shows a code editor with three tabs open:

- parsingXML.py
- new\_products.xml
- new\_products\_updated.xml (selected tab)

The content of new\_products\_updated.xml is as follows:

```
<products>
    <product name="Heineken">
        <id>1</id>
        <price>$1.5</price>
        <slogan formatted="yes">Chỉ Có Thể Là Heineken</slogan>
    </product>
    <product name="Tiger">
        <id>2</id>
        <price>$1.3</price>
        <slogan formatted="yes">Tiger Beer - Bản Lĩnh Đàm Ông</slogan>
    </product>
    <product name="Sapporo">
        <id>3</id>
        <price>$1.4</price>
        <slogan formatted="yes">Tận Hưởng Từng Khoảnh Khắc</slogan>
    </product>
</products>
```

## 2. Xử lý tập tin (file)

### »»» Json file (.json)

```
{  
    "name": "Heineken",  
    "price": "$1.5",  
    "slogan": "....."  
}
```

```
import json  
  
data = {'products': []}  
data['products'].append({  
    'name': 'Heineken',  
    'price': '$1.5',  
    'slogan': 'Chỉ Có Thể Là Heineken'  
})  
data['products'].append({  
    'name': 'Tiger',  
    'price': '$1.3',  
    'slogan': 'Tiger Beer - Bản Lĩnh Đàm Ông'  
})  
data['products'].append({  
    'name': 'Sapporo',  
    'price': '$1.4',  
    'slogan': 'Tận Hưởng Từng Khoảnh Khắc'  
})  
  
with open('data.txt', 'w', encoding='utf8') as outfile:  
    json.dump(data, outfile, ensure_ascii=False)
```

## 2. Xử lý tập tin (file)

### »»» Json file (.json)

```
import json

with open('data.txt', encoding='utf8') as
    json_file:
        data = json.load(json_file)
        for p in data['products']:
            print('Name: ' + p['name'])
            print('Price: ' + p['price'])
            print('Slogan: ' + p['slogan'])
            print()
```

Name: Heineken  
Price: \$1.5  
Slogan: Chỉ Có Thể Là Heineken

Name: Tiger  
Price: \$1.3  
Slogan: Tiger Beer - Bản Lĩnh Đàn Ông

Name: Sapporo  
Price: \$1.4  
Slogan: Tận Hưởng Từng Khoảnh Khắc

## 2. Xử lý tập tin (file)

### »»» Json file (.json)

```
import json
jsonObject = {
    'name': 'Heineken',
    'price': '$1.5',
    'slogan': 'Chỉ Có Thể Là Heineken'
}
jsonString = json.dumps(jsonObject)
print(jsonString)
```

```
{"name": "Heineken", "price": "$1.5", "slogan": "Ch\u01ec9 C\u000f3 Th\u01ec3 L\u000e0 Heineken"}
```

## 2. Xử lý tập tin (file)

### »»» Json file (.json)

```
import json
jsonObject = {
    'name': 'Heineken',
    'price': '$1.5',
    'slogan': 'Chỉ Có Thể Là Heineken'
}
jsonString = json.dumps(jsonObject, ensure_ascii=False)
print(jsonString)
```

```
{"name": "Heineken", "price": "$1.5", "slogan": "Chỉ Có Thể Là Heineken"}
```

## 2. Xử lý tập tin (file)

### »»» Json file (.json)

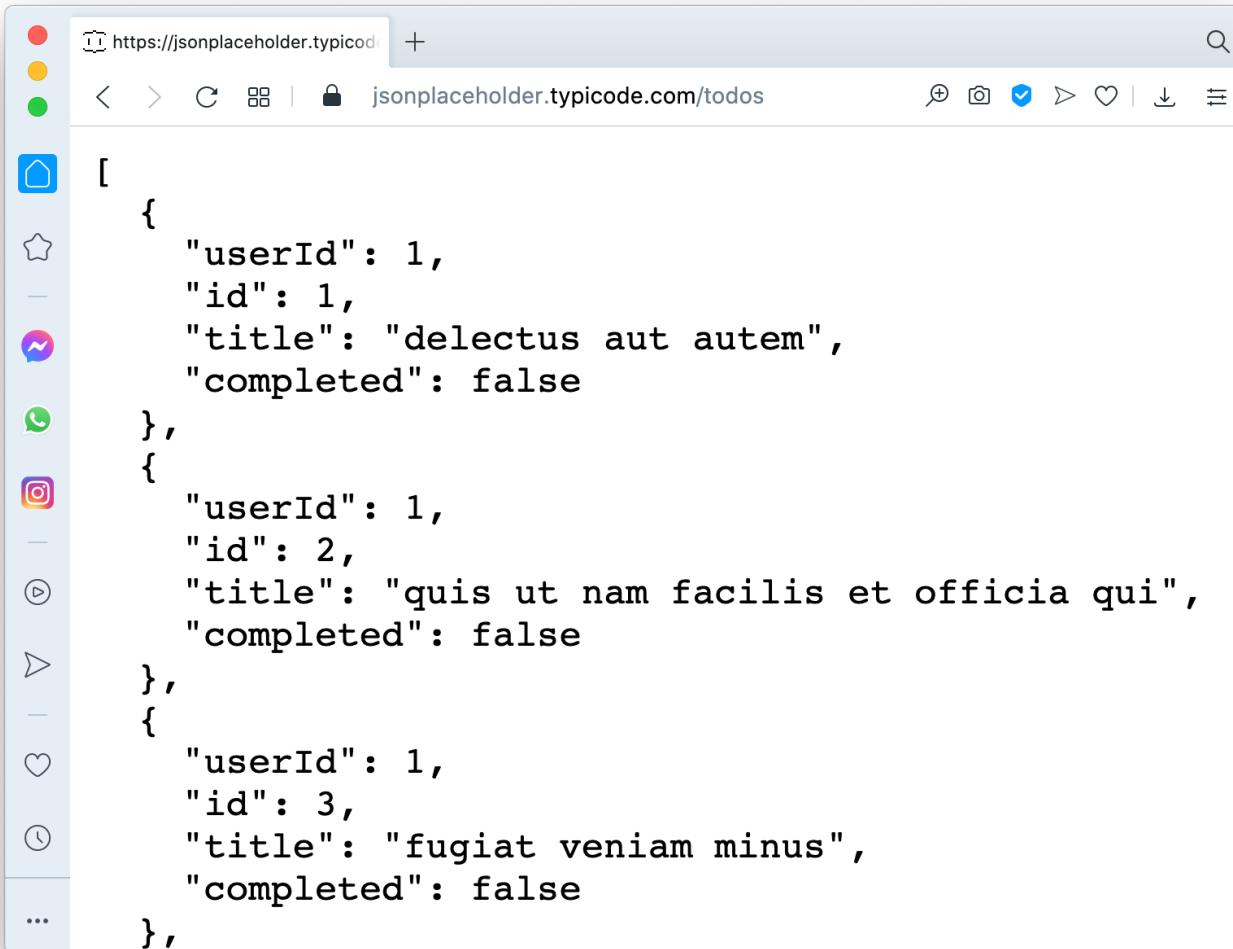
```
import json
jsonObject = {
    'name': 'Heineken',
    'price': '$1.5',
    'slogan': 'Chỉ Có Thể Là Heineken'
}
jsonString = json.dumps(jsonObject, ensure_ascii=False, indent=3)
print(jsonString)
```

```
{
    "name": "Heineken",
    "price": "$1.5",
    "slogan": "Chỉ Có Thể Là Heineken"
}
```

## 2. Xử lý tập tin (file)

### »»» Json file (.json)

#### Đọc dữ liệu Json trả về từ API



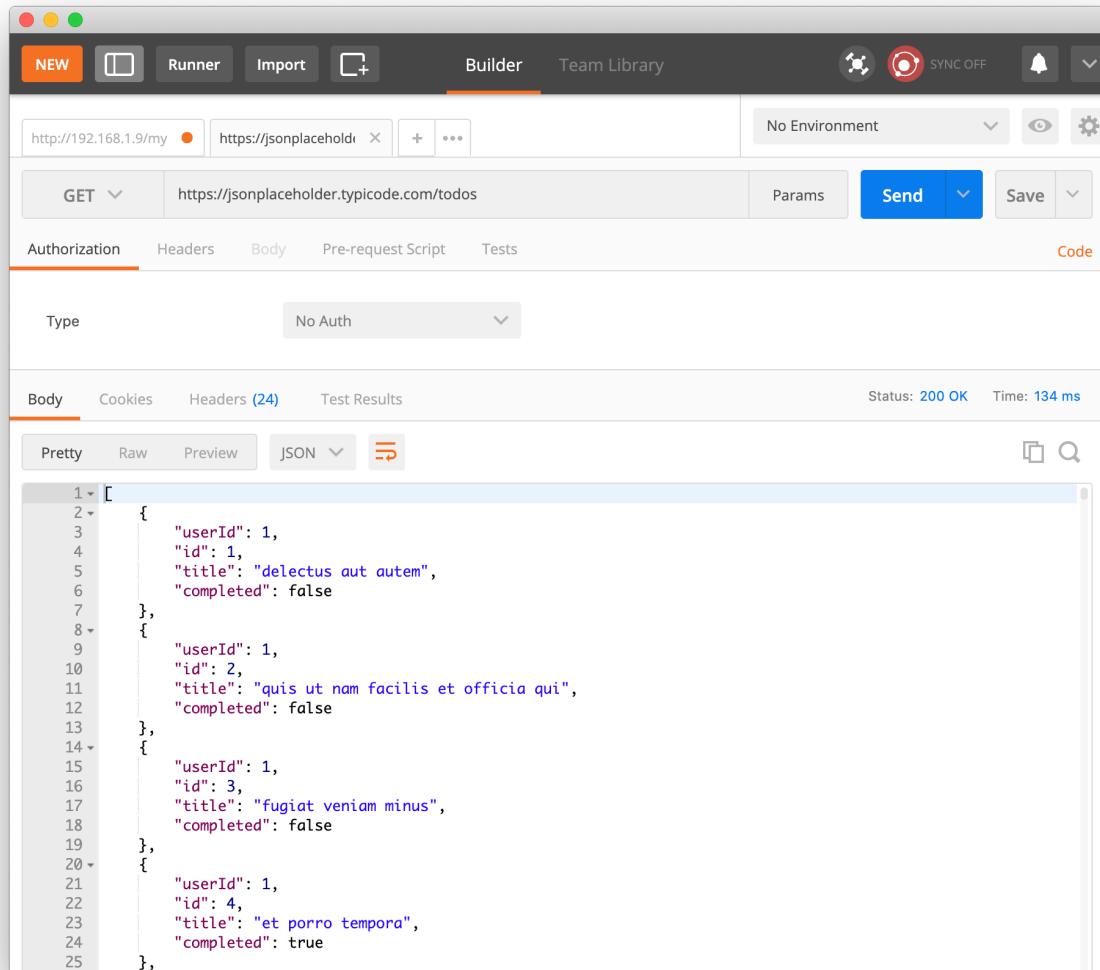
The screenshot shows a web browser window with the URL <https://jsonplaceholder.typicode.com/todos>. The page displays a JSON array of three todo items. Each item is an object with properties: userId, id, title, and completed.

```
[{"userId": 1, "id": 1, "title": "delectus aut autem", "completed": false}, {"userId": 1, "id": 2, "title": "quis ut nam facilis et officia qui", "completed": false}, {"userId": 1, "id": 3, "title": "fugiat veniam minus", "completed": false}...]
```

# 2. Xử lý tập tin (file)

## »»» Json file (.json)

### Đọc dữ liệu Json trả về từ API



## 2. Xử lý tập tin (file)

### »»» Json file (.json)

#### Đọc dữ liệu Json trả về từ API

```
import json
import requests

response = requests.get("https://jsonplaceholder.typicode.com/todos")
# todos = json.loads(response.text)
todos = response.json()
print(todos[0])
for t in todos[:2]:
    jsonObj = json.dumps(t, ensure_ascii=False, indent=3)
    print(jsonObj)
```

## 2. Xử lý tập tin (file)

### »»» Json file (.json)

#### Đọc dữ liệu Json trả về từ API

```
{'userId': 1, 'id': 1, 'title': 'delectus aut autem', 'completed': False}  
{  
    "userId": 1,  
    "id": 1,  
    "title": "delectus aut autem",  
    "completed": false  
}  
{  
    "userId": 1,  
    "id": 2,  
    "title": "quis ut nam facilis et officia qui",  
    "completed": false  
}
```

## 2. Xử lý tập tin (file)

### »»» Xml → Json

```
import xmltodict, json
obj = xmltodict.parse("""
<employee>
    <name>Khánh Hưng</name>
    <role>Lập trình viên</role>
    <age>30</age>
</employee>
""")
print(json.dumps(obj, ensure_ascii=False, indent=3))
```

```
{
    "employee": {
        "name": "Khánh Hưng",
        "role": "Lập trình viên",
        "age": "30"
    }
}
```

## 2. Xử lý tập tin (file)

### Xml → Json

```
import xmltodict, json
```

```
with open("product.xml", "r", encoding="utf8") as f:  
    obj = xmltodict.parse(f.read())  
    print(json.dumps(obj, indent=3))
```

The screenshot shows a code editor window with an XML file named 'product.xml'. The XML structure defines a 'products' container with three nested 'product' elements. Each 'product' element contains an 'id', 'name', and 'price' field. The XML is well-formatted with proper indentation.

```
<?xml version="1.0" encoding="UTF-8"?>  
<products>  
    <product>  
        <id>1</id>  
        <name>Heineken</name>  
        <price>$1.5</price>  
    </product>  
    <product>  
        <id>2</id>  
        <name>Tiger</name>  
        <price>$1.3</price>  
    </product>  
    <product>  
        <id>3</id>  
        <name>Sapporo</name>  
        <price>$1.4</price>  
    </product>  
</products>
```

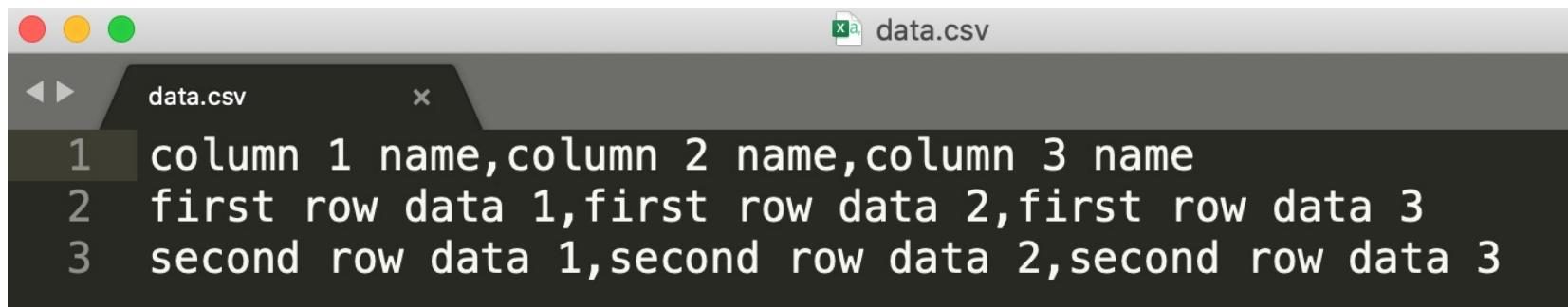
The screenshot shows a code editor window displaying the generated JSON output. The JSON is an object containing a 'products' key, which is itself an object with a 'product' key. This 'product' key points to an array of three objects, each representing a product with its id, name, and price.

```
{  
    "products": {  
        "product": [  
            {  
                "id": "1",  
                "name": "Heineken",  
                "price": "$1.5"  
            },  
            {  
                "id": "2",  
                "name": "Tiger",  
                "price": "$1.3"  
            },  
            {  
                "id": "3",  
                "name": "Sapporo",  
                "price": "$1.4"  
            }  
        ]  
    }  
}
```

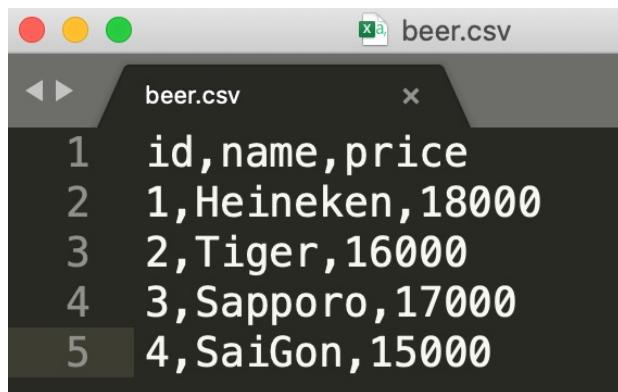
## 2. Xử lý tập tin (file)

»» Excel file (.csv (comma-separated values) / .xlsx)

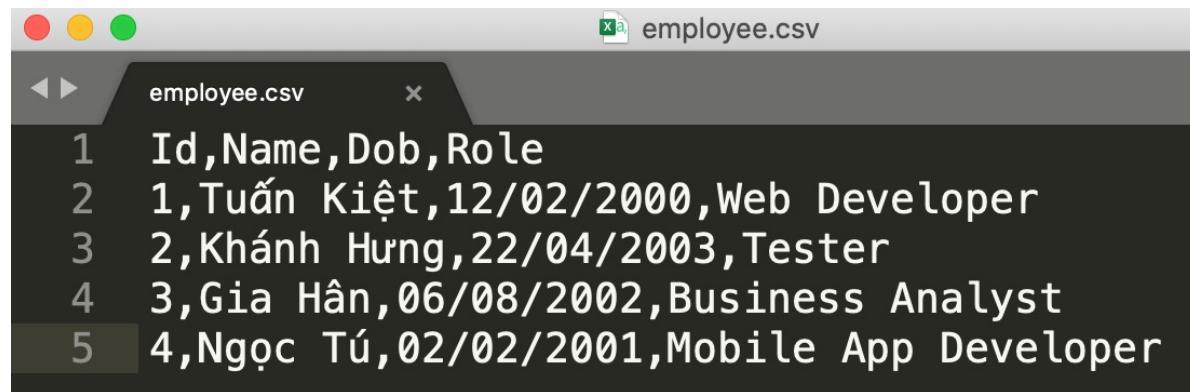
Cấu trúc tập tin .csv



```
data.csv
1 column 1 name,column 2 name,column 3 name
2 first row data 1,first row data 2,first row data 3
3 second row data 1,second row data 2,second row data 3
```



```
beer.csv
1 id,name,price
2 1,Heineken,18000
3 2,Tiger,16000
4 3,Sapporo,17000
5 4,SaiGon,15000
```



```
employee.csv
1 Id,Name,Dob,Role
2 1,TuẤn Kiệt,12/02/2000,Web Developer
3 2,Khánh Hưng,22/04/2003,Tester
4 3,Gia Hân,06/08/2002,Business Analyst
5 4,Ngọc Tú,02/02/2001,Mobile App Developer
```

## 2. Xử lý tập tin (file)

### »» Excel file (.csv (comma-separated values) / .xlsx)

#### Reading .csv file

```
import csv
with open('employee.csv', newline='') as f:
    reader = csv.reader(f, delimiter=',', quoting=csv.QUOTE_NONE)
    for row in reader:
        print(' - '.join(row))
        # print(row[0] + ' * ' + row[1] + ' * ' + row[2] + ' * ' + row[3])
```

Id - Name - Dob - Role

1 - Tuấn Kiệt - 12/02/2000 - Web Developer

2 - Khánh Hưng - 22/04/2003 - Tester

3 - Gia Hân - 06/08/2002 - Business Analyst

4 - Ngọc Tú - 02/02/2001 - Mobile App Developer

## 2. Xử lý tập tin (file)

»» Excel file (.csv (comma-separated values) / .xlsx)

Reading .csv file (using pandas library)

```
import pandas  
df = pandas.read_csv('employee.csv')  
print(df)
```

	<b>Id</b>	<b>Name</b>	<b>Dob</b>	<b>Role</b>
0	1	Tuấn Kiệt	12/02/2000	Web Developer
1	2	Khánh Hưng	22/04/2003	Tester
2	3	Gia Hân	06/08/2002	Business Analyst
3	4	Ngọc Tú	02/02/2001	Mobile App Developer

## 2. Xử lý tập tin (file)

### »» Excel file (.csv (comma-separated values) / .xlsx)

#### Writing .csv file

```
import csv

with open('new_employee.csv', mode='w') as f:
    employee_writer = csv.writer(f, delimiter=',', quotechar='"')
    employee_writer.writerow(['Id', 'Name', 'Dob'])
    employee_writer.writerow(['1', 'Tú Linh', '02/02/2002'])
    employee_writer.writerow(['2', 'Nam Giao', '03/04/2000'])
    employee_writer.writerow(['3', 'Huỳnh Anh', '05/11/2001'])
```

The screenshot shows a code editor with two tabs: 'read\_write\_csv.py' and 'new\_employee.csv'. The Python script 'read\_write\_csv.py' contains the code provided in the previous block. The 'new\_employee.csv' tab shows the generated CSV file with the following data:

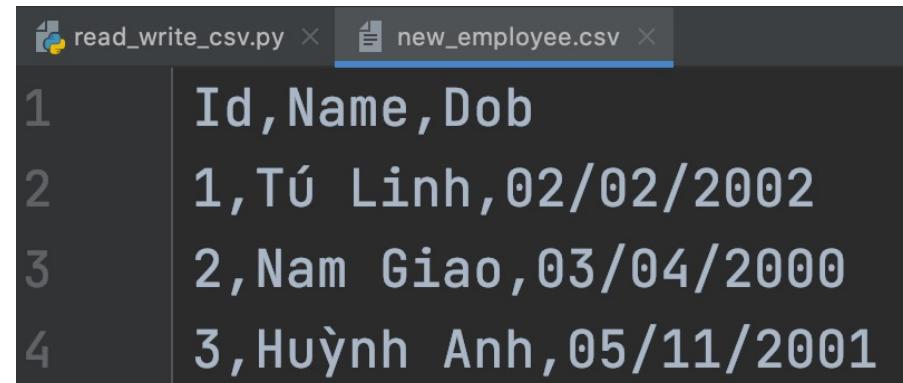
	Id, Name, Dob
1	1, Tú Linh, 02/02/2002
2	2, Nam Giao, 03/04/2000
3	3, Huỳnh Anh, 05/11/2001

## 2. Xử lý tập tin (file)

### »» Excel file (.csv (comma-separated values) / .xlsx)

#### Writing .csv file

```
import csv
with open('new_employee.csv', mode='w') as f:
    fieldnames = ['Id', 'Name', 'Dob']
    writer = csv.DictWriter(f, fieldnames=fieldnames)
    writer.writeheader()
    writer.writerow({'Id': '1', 'Name': 'Tú Linh', 'Dob': '02/02/2002'})
    writer.writerow({'Id': '2', 'Name': 'Nam Giao', 'Dob': '03/04/2000'})
    writer.writerow({'Id': '3', 'Name': 'Huỳnh Anh', 'Dob': '05/11/2001'})
```



The screenshot shows a terminal window with two tabs: 'read\_write\_csv.py' and 'new\_employee.csv'. The 'new\_employee.csv' tab is active, displaying the following data:

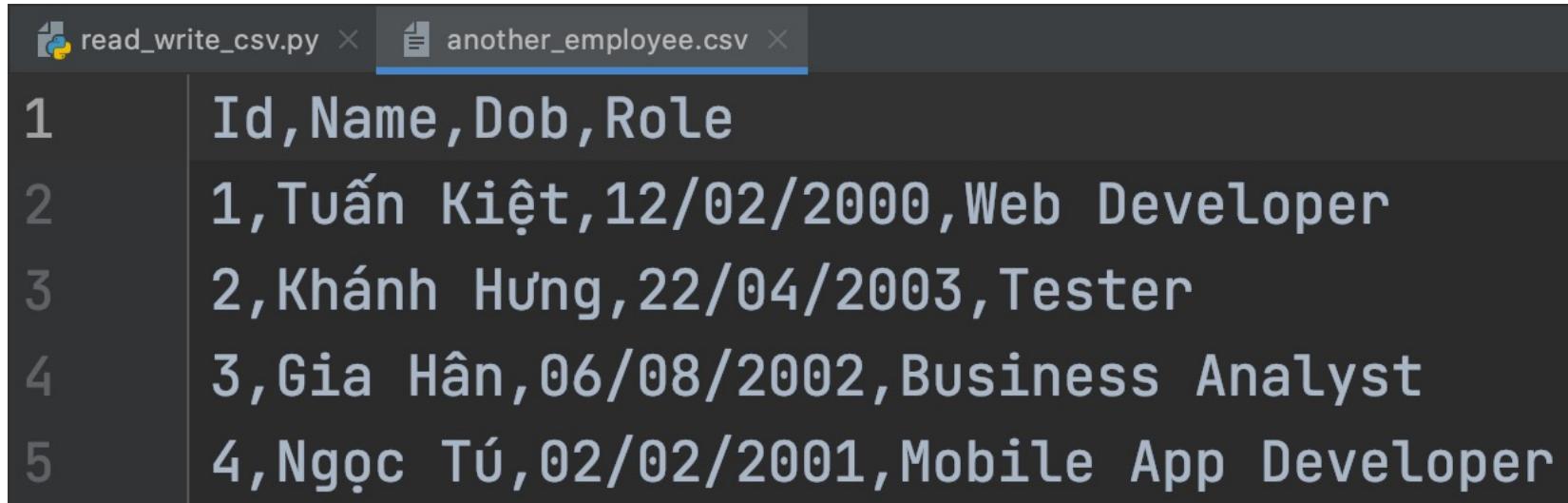
1	Id,Name,Dob
2	1,Tú Linh,02/02/2002
3	2,Nam Giao,03/04/2000
4	3,Huỳnh Anh,05/11/2001

## 2. Xử lý tập tin (file)

### »» Excel file (.csv (comma-separated values) / .xlsx)

#### Writing .csv file (using pandas library)

```
import pandas  
df = pandas.read_csv('employee.csv', index_col="Id")  
df.to_csv('another_employee.csv')
```



	Id, Name, Dob, Role
1	1, Tuấn Kiệt, 12/02/2000, Web Developer
2	2, Khánh Hưng, 22/04/2003, Tester
3	3, Gia Hân, 06/08/2002, Business Analyst
4	4, Ngọc Tú, 02/02/2001, Mobile App Developer

## 2. Xử lý tập tin (file)

### »» Excel file (.csv (comma-separated values) / .xlsx)

#### Writing .xlsx file

```
import xlsxwriter as xr

workbook = xr.Workbook("Products.xlsx")
worksheet = workbook.add_worksheet()

# Modify column width
worksheet.set_column('A:A', 5)
worksheet.set_column('B:B', 20)
worksheet.set_column('C:C', 15)

bold = workbook.add_format({'bold': True})

# Add header
worksheet.write('A1', 'Id', bold)
worksheet.write('B1', 'Name', bold)
worksheet.write('C1', 'Price', bold)
```

```
# Add first row
worksheet.write('A2', '1')
worksheet.write('B2', 'Heineken')
worksheet.write('C2', '19000')

# Add second row
worksheet.write('A3', '2')
worksheet.write('B3', 'Tiger')
worksheet.write('C3', '18000')

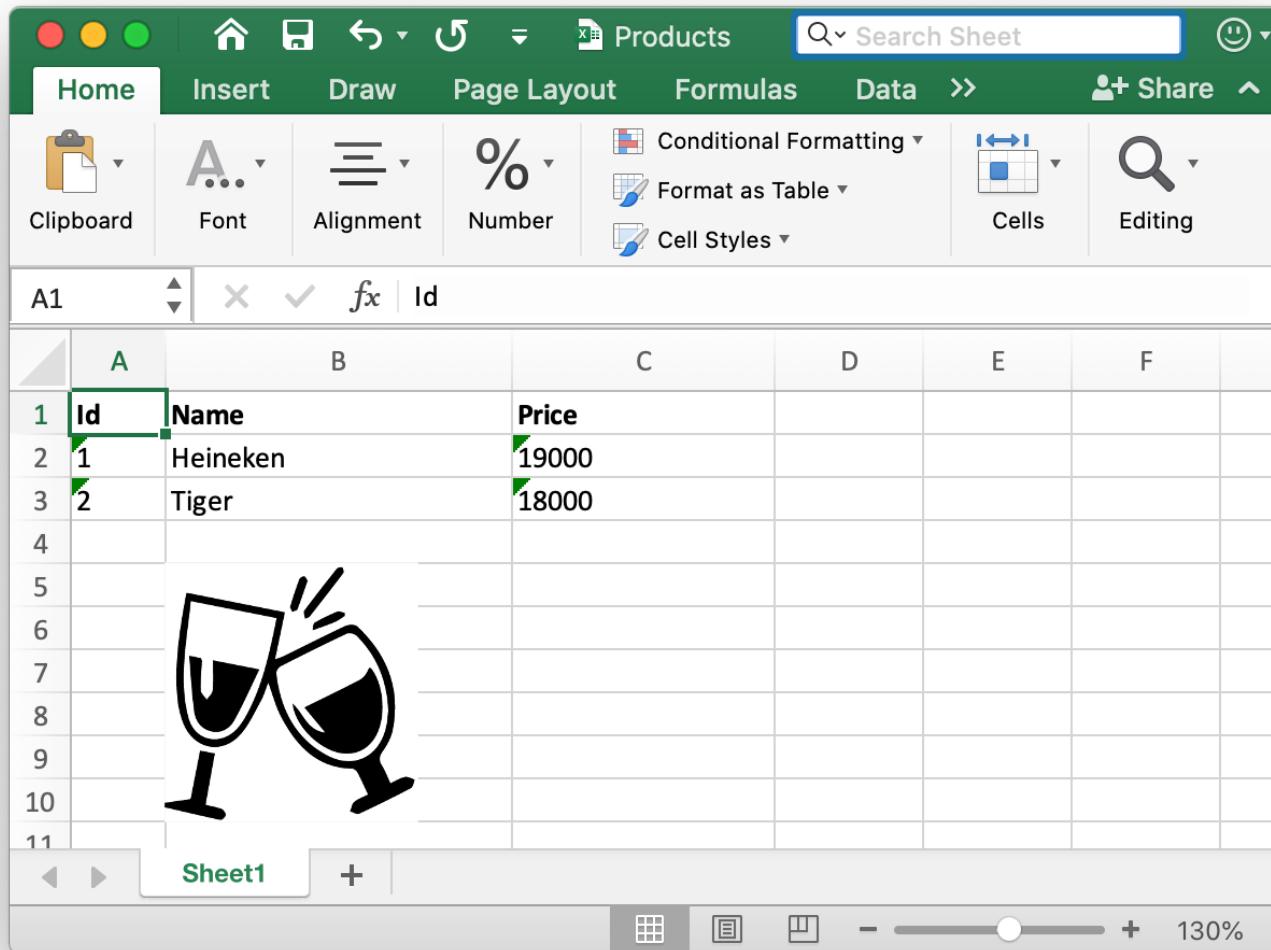
# Insert image
worksheet.insert_image('B5', 'beer.png')

workbook.close()
```

## 2. Xử lý tập tin (file)

## »»» Excel file (.csv (comma-separated values) / .xlsx)

## Writing .xlsx file



## 2. Xử lý tập tin (file)

### »» Excel file (.csv (comma-separated values) / .xlsx)

#### Reading .xlsx file

```
from openpyxl import load_workbook  
wb = load_workbook('Products.xlsx')  
print(wb.sheetnames)  
ws = wb[wb.sheetnames[0]]  
for row in ws.values:  
    for value in row:  
        print(value.center(9), end="")  
    print("")
```

['Sheet1']		
	Name	Price
1	Heineken	19000
2	Tiger	18000

## 2. Xử lý tập tin (file)

»» Excel file (.csv (comma-separated values) / .xlsx)

### Reading .xlsx file

\$ pip install xlrld → .xls

\$ pip install openpyxl → .xlsx

```
import pandas as pd  
df = pd.read_excel('Sales.xlsx')
```

	Week	Sales_Volume	Price	Ads_Cost
0	1	350	5.50000	3.30000
1	2	460	7.50000	3.30000
2	3	350	8.00000	3.00000
3	4	430	8.00000	4.50000
4	5	350	6.80000	3.00000

## 2. Xử lý tập tin (file)

### »»» Multimedia file (.jpg, .png, .mp3, .mp4, ...)

#### Read images

```
from PIL import Image
```

```
# Read image  
img = Image.open('python.png')
```

```
# Output Images  
img.show()
```

```
# Prints format of image  
print(img.format) PNG
```

```
# Prints mode of image  
print(img.mode) RGBA
```



# Q & A