

PYTHON

FILE

Tien-Lam Pham
Anh-Tuan Nguyen
Phenikaa School of Computing

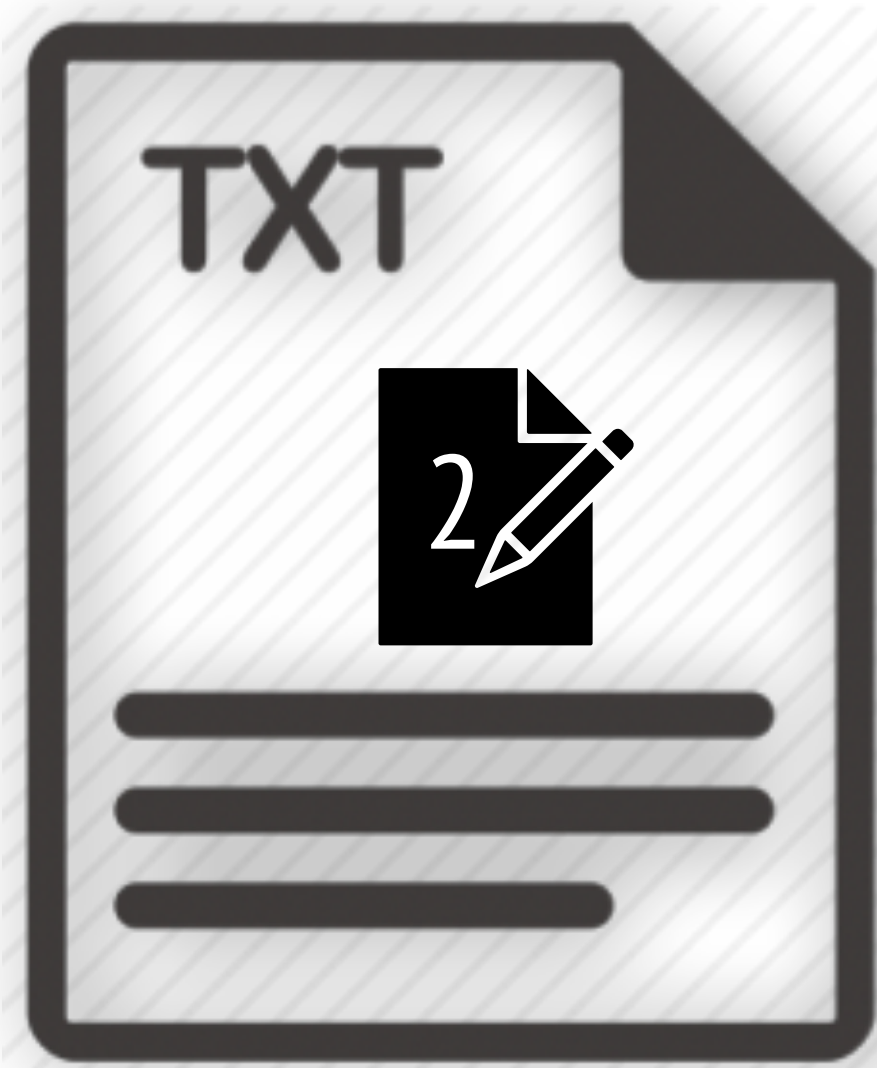
OUTLINE

- procedure
- working with file by pandas

TYPICAL PROCEDURE

Python

1 ↓ ↑ 3



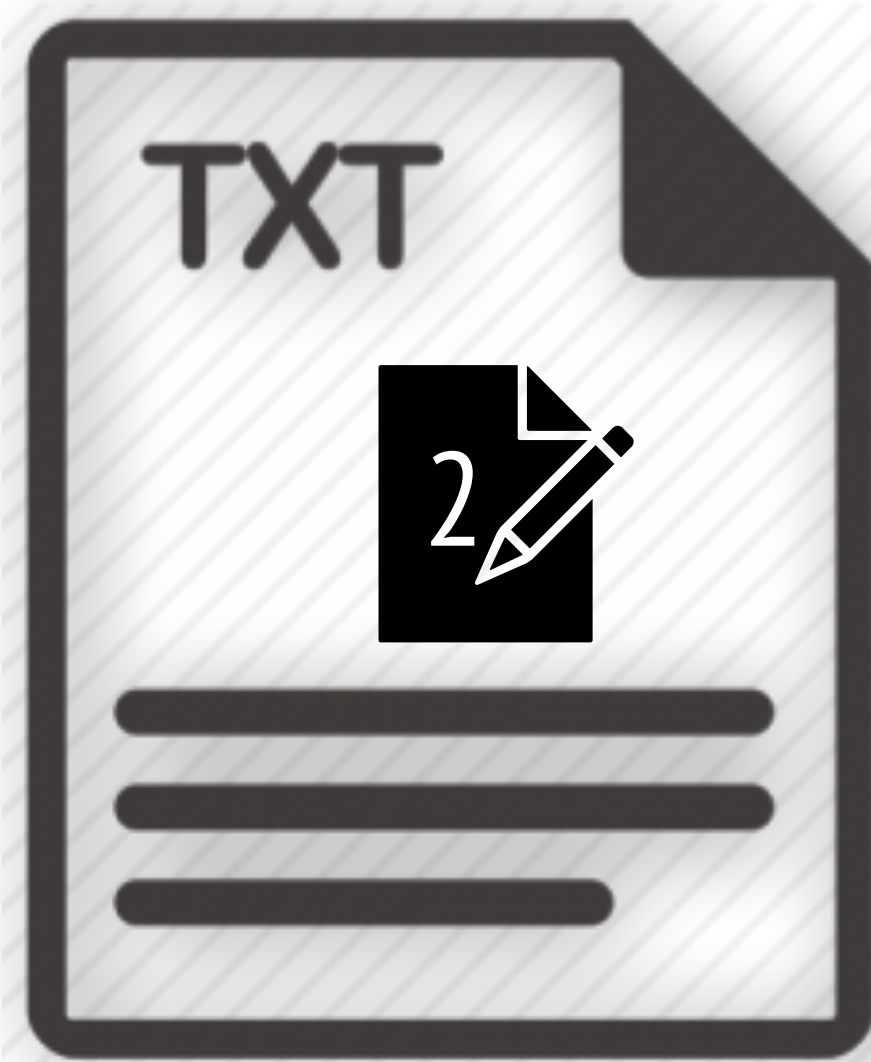
- ➡ 1. Connect to file
- ➡ 2. Read from/ write to file
- ➡ 3. Disconnect to file

READ FROM A FILE

- file already exists

Python

1 ↓ ↑ 3



→ `open(file_path, 'r')`

→ `read()`

→ `close()`

```
file.py
1 # connect with a file
2 file = open("example.txt", 'r')
3
4 # read content as string
5 data = file.read()
6
7 print(type(data))
8 print(data)
9
10 # disconnect with a file
11 file.close()
```

Line 11, Column 13

```
example.txt
1 My name is Quyen
2 Hello
3 How are you today?
```

Column 19 Tab Size: 4

```
ex — -zsh — 53x13
[(base) mac@Quyens-MacBook-Pro ex % python file.py
<class 'str'>
My name is Quyen
Hello
How are you today?
(base) mac@Quyens-MacBook-Pro ex %
```


READ FROM A FILE

- file already exists, read content from a file as lines

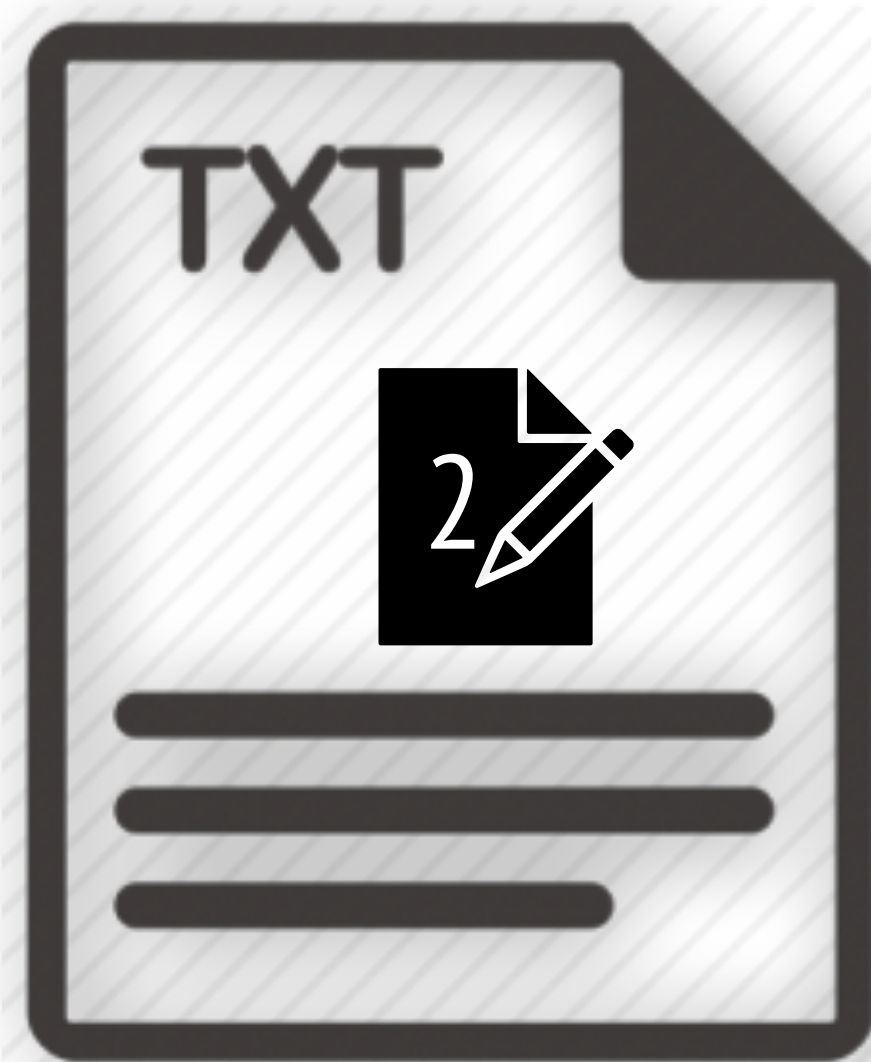
Python

→ `open(file_path, 'r')`

→ `readlines()`

→ `close()`

1 ↓ ↑ 3



```
file.py
example.txt x
1 # connect with a file
2 file = open("example.txt", 'r')
3
4 # read content as string
5 lines = file.readlines()
6
7 for line in lines:
8     print(line)
9
10 # disconnect with a file
11 file.close()
```

Line 11, Column 13

```
example.txt UNREGISTERED
example.txt x
1 My name is Quyen
2 Hello
3 How are you today?
```

Column 19 Tab Size: 4

```
ex — -zsh — 53x13
[(base) mac@Quyens-MacBook-Pro ex % python file.py
My name is Quyen

Hello

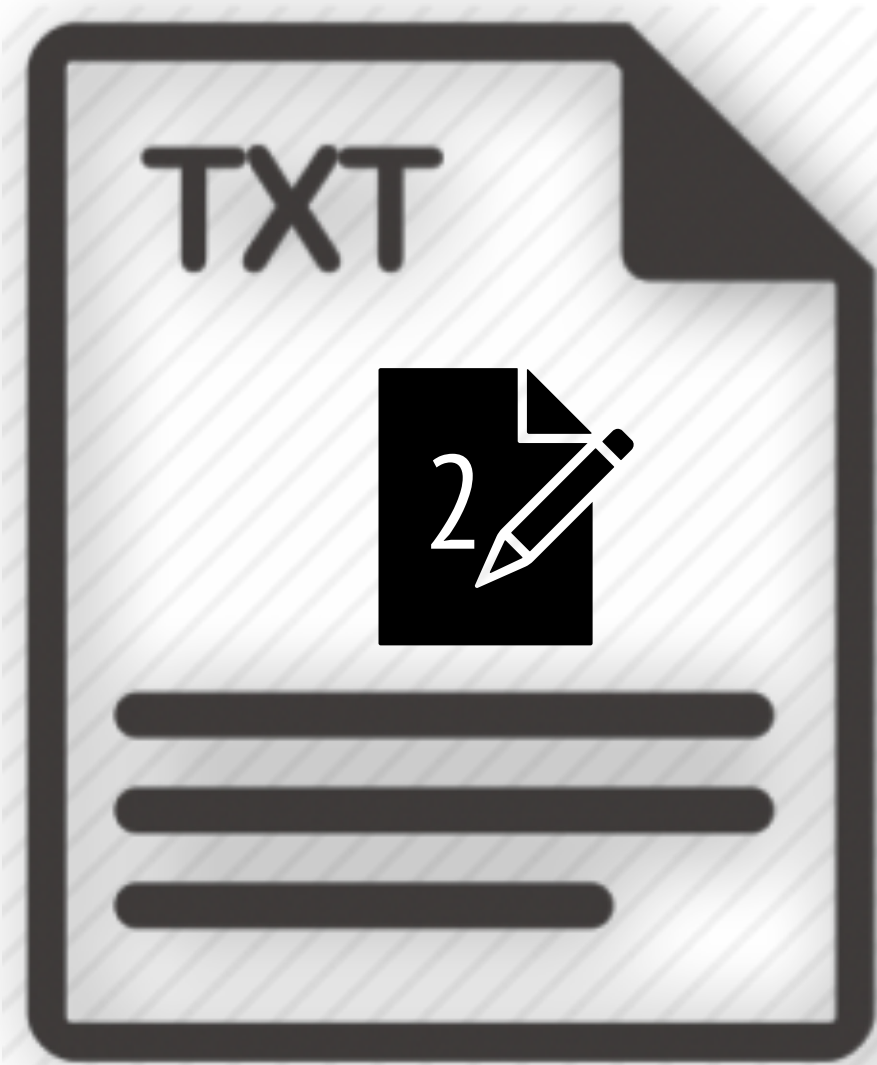
How are you today?
(base) mac@Quyens-MacBook-Pro ex %
```

WRITE TO A FILE

- file does not exist

Python

1 ↓ ↑ 3



→ `open(file_path, 'w')`

→ `write()`

→ `close()`

```
newfile.py  UNREGISTERED
1  # connect with a file
2  file = open("new_file.txt", 'w')
3
4  line1 = "this is line 1 \n"
5  line2 = "this is line 2 \n"
6
7  file.write(line1)
8  file.write(line2)
9
10 # disconnect with a file
11 file.close()
```

Line 11, Column 13

```
new_file.txt  UNREGISTERED
1  this is line 1
2  this is line 2
3
```

Line 3, Column 1 Tab Size: 4

WRITE TO A FILE

- file already exists (appending content if the file is exist)

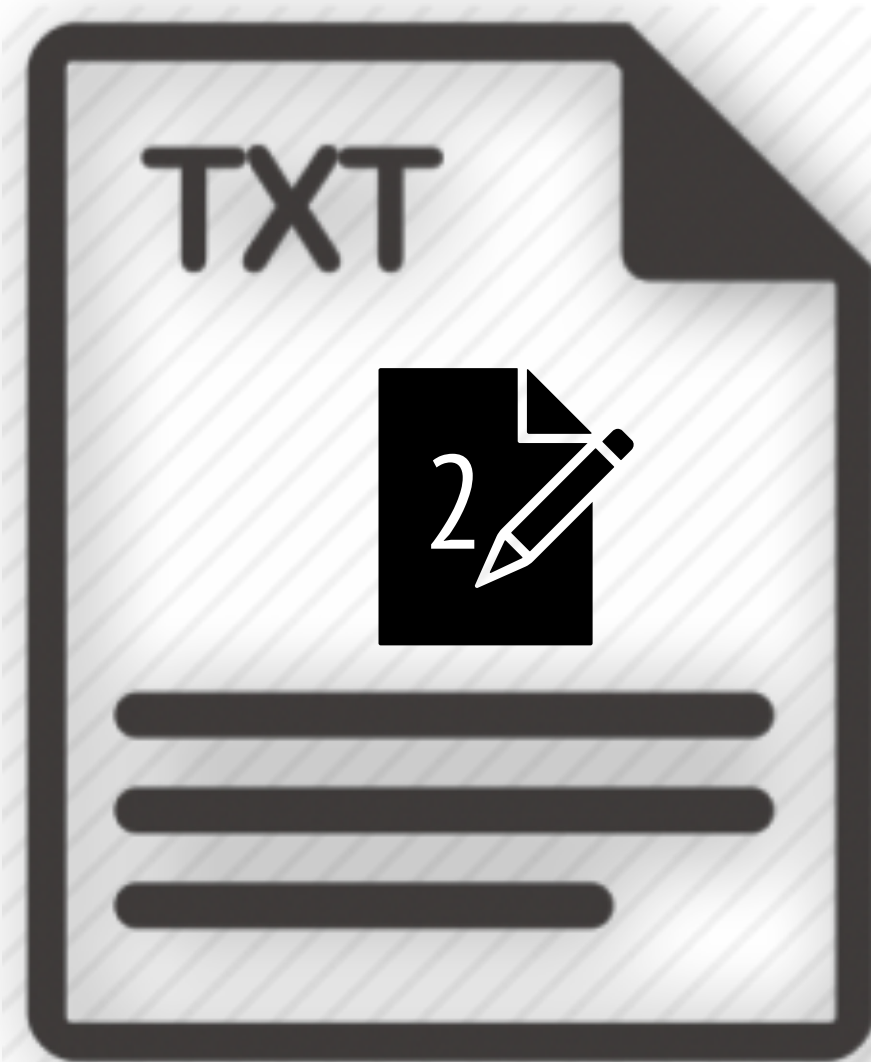
Python

→ `open(file_path, 'a')`

→ `write()`

→ `close()`

1 ↓ ↑ 3



```
append_line.py  UNREGISTERED
1 # connect with a file
2 file = open("new_file.txt", 'a')
3
4 line3 = "this is line 3 \n"
5
6 file.write(line3)
7
8 # disconnect with a file
9 file.close()
```

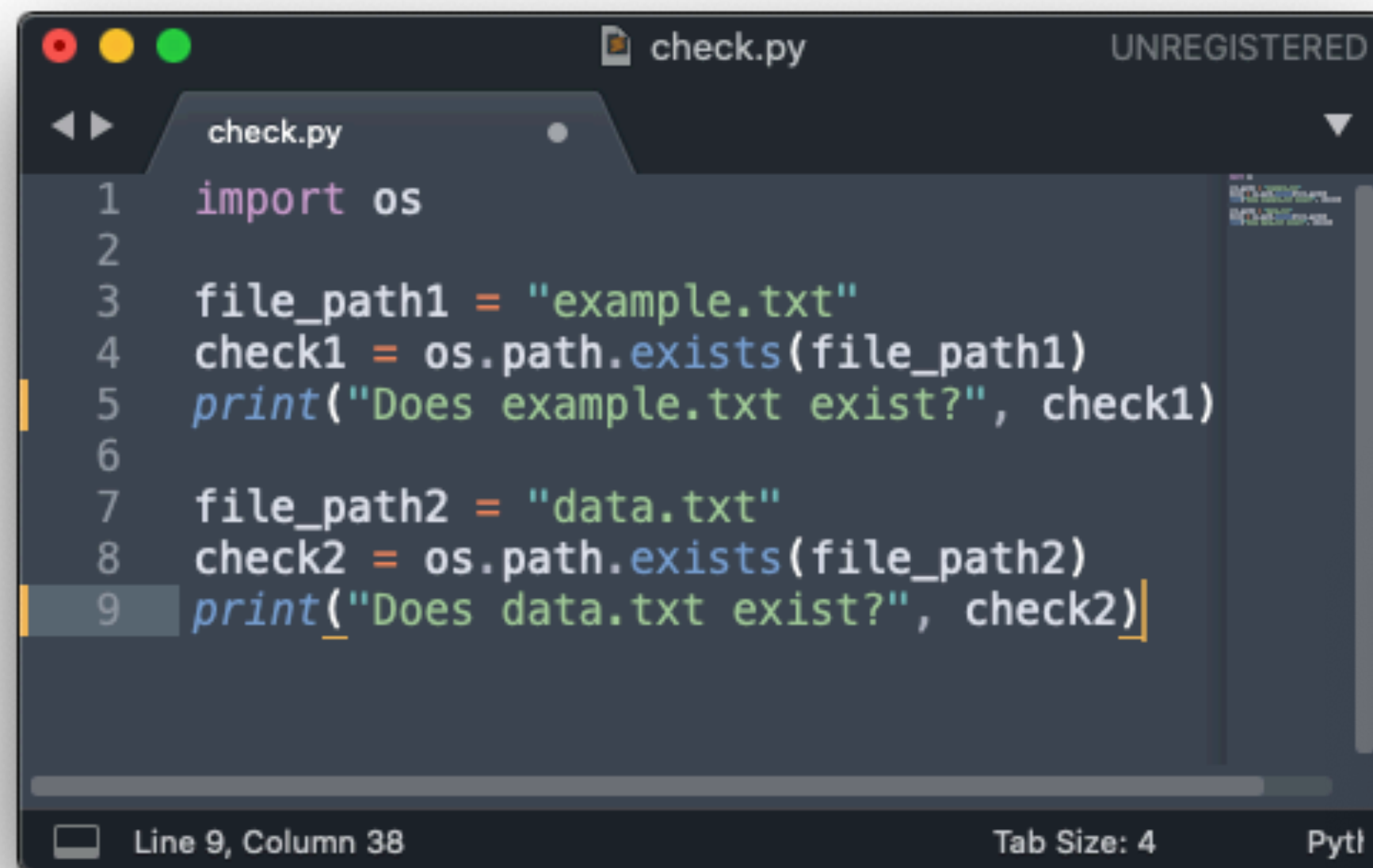
Line 9, Column 13

```
new_file.txt  UNREGISTERED
1 this is line 1
2 this is line 2
3 this is line 3
4
```

Line 4, Column 1; Reloading ~/Dropbox/Lam-Quyen/HUS/python_progr:

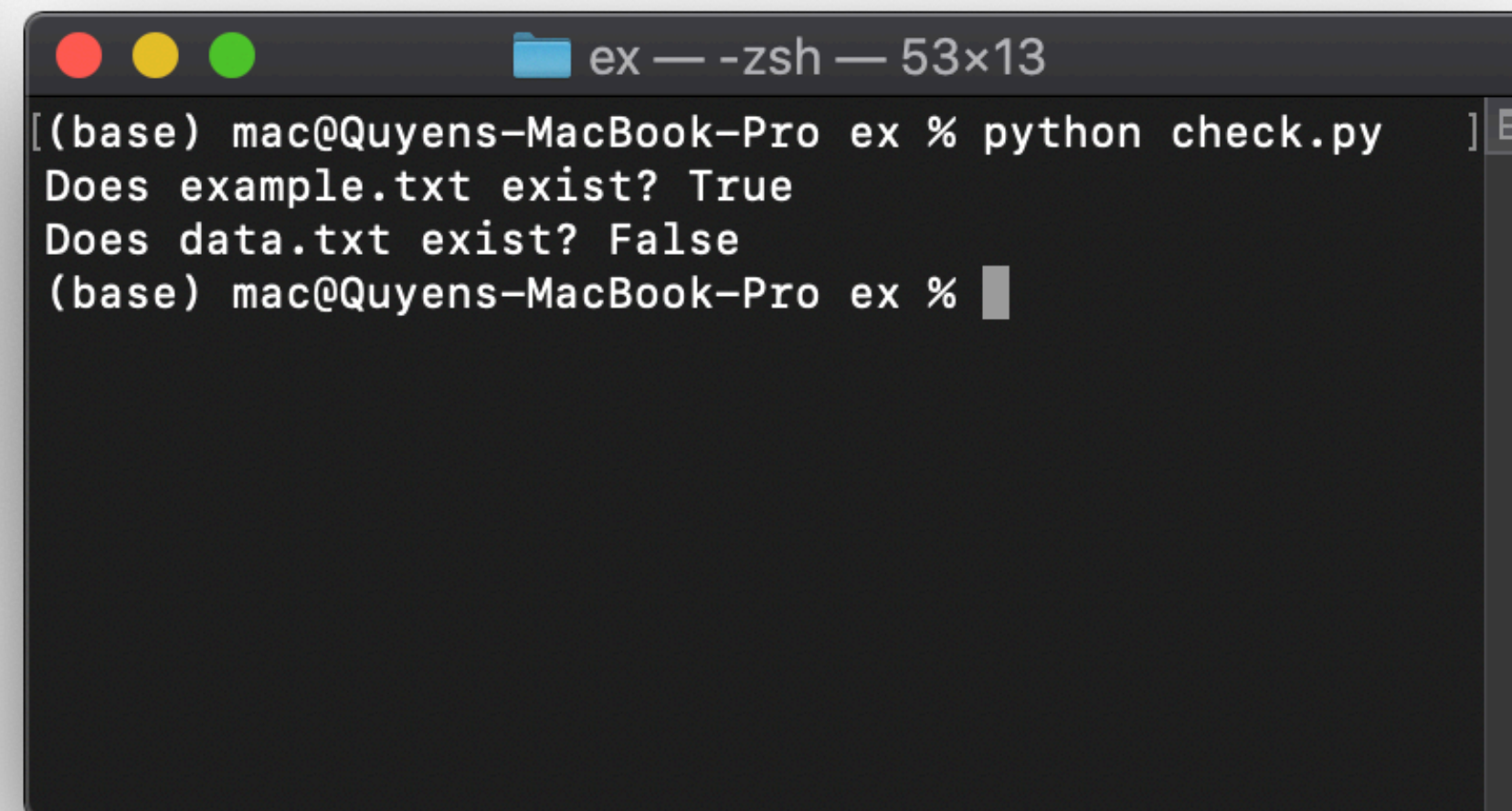
USEFUL FUNCTIONS

- Check if a file exists



```
1 import os
2
3 file_path1 = "example.txt"
4 check1 = os.path.exists(file_path1)
5 print("Does example.txt exist?", check1)
6
7 file_path2 = "data.txt"
8 check2 = os.path.exists(file_path2)
9 print("Does data.txt exist?", check2)
```

The screenshot shows a code editor window titled 'check.py' with a dark theme. The code is a Python script that checks the existence of two files: 'example.txt' and 'data.txt'. The script uses the `os.path.exists()` function. The status bar at the bottom indicates 'Line 9, Column 38', 'Tab Size: 4', and 'Pytl'.

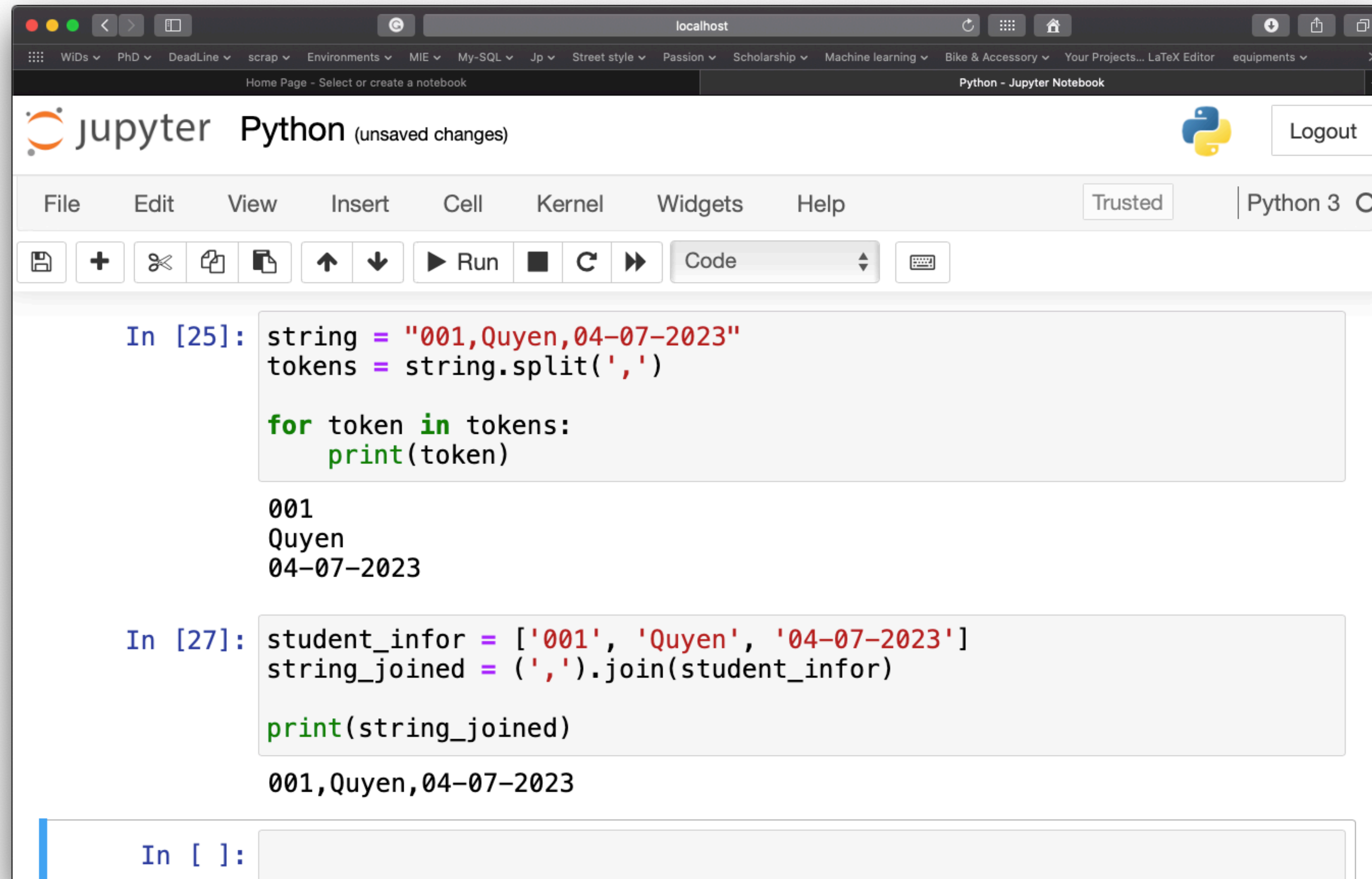


```
ex — -zsh — 53x13
(base) mac@Quyens-MacBook-Pro ex % python check.py
Does example.txt exist? True
Does data.txt exist? False
(base) mac@Quyens-MacBook-Pro ex %
```

The screenshot shows a terminal window titled 'ex — -zsh — 53x13'. It displays the output of running the `python check.py` command. The output shows that 'example.txt' exists (True) and 'data.txt' does not exist (False). The prompt is '(base) mac@Quyens-MacBook-Pro ex %'.

USEFUL FUNCTIONS

- String splitting/ String joining



The screenshot shows a Jupyter Notebook interface with a dark-themed browser window. The notebook is titled 'Python (unsaved changes)' and has a 'Logout' button. The menu bar includes 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. The toolbar contains icons for saving, adding cells, cutting, copying, pasting, undo, redo, and running code. The code area shows two input cells. The first cell, labeled 'In [25]:', contains Python code that splits a string '001,Quyen,04-07-2023' into a list of tokens and prints each token on a new line. The second cell, labeled 'In [27]:', contains Python code that joins a list of tokens back into a single string and prints it. The output of the first cell is '001', 'Quyen', and '04-07-2023' on separate lines. The output of the second cell is '001,Quyen,04-07-2023'.

```
In [25]: string = "001,Quyen,04-07-2023"
tokens = string.split(',')

for token in tokens:
    print(token)

001
Quyen
04-07-2023

In [27]: student_infor = ['001', 'Quyen', '04-07-2023']
string_joined = ('').join(student_infor)

print(string_joined)

001,Quyen,04-07-2023

In [ ]:
```

WORKING WITH FILE BY PANDAS



- read csv, excel, text, ...
- export to csv, excel, ...

- ➡ 1. Connect to file
- ➡ 2. Read from/ write to file
- ➡ 3. Disconnect to file

ASSIGNMENT
