

## Lab 12 Summary

Jhonatan Parada

ET574

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS ... bash - Jhonatan_LAB12

@jhonatanparada499 → /workspaces/ET574/labs/Jhonatan_LAB12 (main) $ ls -R
.:
data lab12_1.py lab12_2.py lab12_3.py modules script.py

./data:
Presidents.txt

./modules:
file.py

@jhonatanparada499 → /workspaces/ET574/labs/Jhonatan_LAB12 (main) $

labs > Jhonatan_LAB12 > lab12_1.py > ...
1 # lab12_1.py - Jhonatan Parada
2
3 def print_file_content(file_path, mode):
4     if not isinstance(file_path, str):
5         print(f'{file_path} must be a string'); return
6
7     if not isinstance(mode, str):
8         print(f'{mode} must be a string'); return
9
10    try: file_obj = open(file_path)
11    except FileNotFoundError:
12        print('file path not found'); return
13
14    modes = ('loop', 'list')
15    mode = mode.lower()
16
17    if not mode in modes:
18        print(f'{mode} is not a valid mode')
19        print(f'valid modes: {modes}')
20        file_obj.close()
21    return
22
23    # print the contents using loop or list
24    if mode == 'loop':
25        for line in file_obj:
26            print(line.strip())
27
28    elif mode == 'list':
```

```

28     elif mode == 'list':
29         print(*file_obj.readlines(), sep='')
30
31     file_obj.close()
32
33 def main():
34     # .sh (Unix, bash)
35     my_file_path = 'data//Presidents.txt'
36
37     # .bat (Windows, cmd, powershell )
38     # my_file_path = 'data\\Presidents.txt'
39
40     print('Using Loop:')
41     print_file_content(my_file_path, 'loop')
42     print('\nUsing List:')
43     print_file_content(my_file_path, 'list')
44
45 if __name__ == '__main__': main()

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```

● @jhonatanparada499 → /workspaces/ET574/labs/Jhonatan_LAB12 (main) $ python lab12_1.py
Using Loop:
George Washington
John Adams
Thomas Jefferson

Using List:
George Washington
John Adams
Thomas Jefferson
○ @jhonatanparada499 → /workspaces/ET574/labs/Jhonatan_LAB12 (main) $ █

```

labs > Jhonatan\_LAB12 > lab12\_2.py > create\_file

```
1  # lab12_2.py - Jhonatan Parada
2
3  def create_file(full_name, content=''):
4      if not isinstance(full_name, str):
5          print(f'{full_name} must be a string')
6          return
7
8      if not isinstance(content, (list, tuple, str)):
9          print(f'{content} must be a string or iterable')
10         return
11
12     if isinstance(content, (list, tuple)):
13         content = '\n'.join(content)
14
15     try:
16         new_file = open(full_name, 'x')
17     except FileNotFoundError:
18         print(f'{full_name} not found'); return
19     except FileExistsError:
20         print(f'{full_name} already exists'); return
21
22     new_file.write(content)
23     new_file.close
24
25 def main():
26     # .sh (Unix, bash)
27     new_file_1 = 'data//Presidents2.txt'
28     # .sh (Unix, bash)
```

```
Python 3.10.0 Shell
lab12_2.py
def create_file(full_name, content=''):
    if not isinstance(full_name, str):
        print(f'{full_name} must be a string')
        return
    if not isinstance(content, (list, tuple, str)):
        print(f'{content} must be a string or iterable')
        return
    if isinstance(content, (list, tuple)):
        content = '\n'.join(content)
    try:
        new_file = open(full_name, 'x')
    except FileNotFoundError:
        print(f'{full_name} not found'); return
    except FileExistsError:
        print(f'{full_name} already exists'); return
    new_file.write(content)
    new_file.close
def main():
    # .sh (Unix, bash)
    new_file_1 = 'data//Presidents2.txt'
    # .sh (Unix, bash)
```

```

28 # .sh (Unix, bash)
29 new_file_2 = 'data//Presidents3.txt'
30
31 # .bat (Windows, cmd, powershell )
32 # my_file_1 = 'data\\Presidents2.txt'
33
34 # .bat (Windows, cmd, powershell )
35 # my_file_2 = 'data\\Presidents3.txt'
36
37 presidents = (
38     'George Washington',
39     'John Adams',
40     'Thomas Jefferson'
41 )
42
43 create_file(new_file_1, presidents[:2])
44 create_file(new_file_2, presidents)
45
46 if __name__ == '__main__': main()

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```

• @jhonatanparada499 →/workspaces/ET574/labs/Jhonatan_LAB12 (main) $ python lab12_2.py
• @jhonatanparada499 →/workspaces/ET574/labs/Jhonatan_LAB12 (main) $ ls -R
.:
data lab12_1.py lab12_2.py lab12_3.py modules script.py

./data:
Presidents.txt Presidents2.txt Presidents3.txt

./modules:
file.py
• @jhonatanparada499 →/workspaces/ET574/labs/Jhonatan_LAB12 (main) $ cat data/Presidents2.txt
George Washington
• John Adams@jhonatanparada499 →/workspaces/ET574/labs/Jhonatan_LAB12 (main) $ cat data/Presidents3.txt
George Washington
John Adams
○ Thomas Jefferson@jhonatanparada499 →/workspaces/ET574/labs/Jhonatan_LAB12 (main) $ █

```

labs > Jhonatan\_LAB12 > lab12\_3.py > modify\_file

```
1  # lab12_3.py - Jhonatan Parada
2
3  def modify_file(file_target, action, content=''):
4      if not isinstance(file_target, str):
5          print(f'{file_target} must be a string')
6          return
7
8      if not isinstance(action, str):
9          print(f'{action} must be a string')
10         return
11
12     if not isinstance(content, (list, tuple, str)):
13         print(f'{content} must be a string or iterable')
14         return
15
16     if isinstance(content, (list, tuple)):
17         content = '\n'.join(content)
18
19     actions = ('write', 'append')
20     action = action.lower()
21
22     if not action in actions:
23         print(f'{action} is not a valid action')
24         print(f'valid actions: {actions}')
25         return
26
27     if action == 'append':
28
29         content = '\n' + content
30
31     try:
32         file_target = open(file_target, action[0])
33     except FileNotFoundError:
34         print(f'{file_target} not found'); return
35
36     file_target.write(content)
37     file_target.close()
```

```
"""
    This module contains the modify_file function, which
    allows to modify a file. It takes three arguments:
    file_target (str), action (str), and content (str or
    iterable). It returns None.
"""
```

```
"""
    This module contains the modify_file function, which
    allows to modify a file. It takes three arguments:
    file_target (str), action (str), and content (str or
    iterable). It returns None.
"""
```

labs > Jhonatan\_LAB12 > modules > file.py

```
1 # lab12_3.py (module part)
2 from lab12_1 import print_file_content as print_content
3 from lab12_2 import create_file as create
4 from lab12_3 import modify_file as modify
```

labs > Jhonatan\_LAB12 > script.py > ...

```
1 # lab12_3.py (script part)
2 from modules import file
3
4 def main():
5     # .sh (Unix, bash)
6     my_file_target = 'data//Presidents4.txt'
7
8     # .bat (Windows, cmd, powershell )
9     # my_file_target = 'data\\Presidents4.txt'
10
11     presidents = (
12         'George Washington',
13         'John Adams',
14         'Thomas Jefferson',
15         'James Madison',
16         'James Monroe',
17         'John Quincy Adams'
18     )
19
20     file.modify(my_file_target, 'write', presidents[:3])
21     file.modify(my_file_target, 'append', presidents[3:])
22     file.print_content(my_file_target, 'loop')
23
24 if __name__ == '__main__': main()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

- @jhonatanparada499 → /workspaces/ET574/labs/Jhonatan\_LAB12 (main) \$ python script.py  
George Washington  
John Adams  
Thomas Jefferson  
James Madison  
James Monroe  
John Quincy Adams
- @jhonatanparada499 → /workspaces/ET574/labs/Jhonatan\_LAB12 (main) \$ ls -R  
.:  
\_\_pycache\_\_ data lab12\_1.py lab12\_2.py lab12\_3.py modules script.py  
  
./\_\_pycache\_\_:  
lab12\_1.cpython-312.pyc lab12\_2.cpython-312.pyc lab12\_3.cpython-312.pyc  
  
./data:  
Presidents.txt Presidents2.txt Presidents3.txt Presidents4.txt  
  
./modules:  
\_\_pycache\_\_ file.py  
  
./modules/\_\_pycache\_\_:  
file.cpython-312.pyc
- @jhonatanparada499 → /workspaces/ET574/labs/Jhonatan\_LAB12 (main) \$ cat data/Presidents4.txt  
George Washington  
John Adams  
Thomas Jefferson  
James Madison  
James Monroe
- John Quincy Adams @jhonatanparada499 → /workspaces/ET574/labs/Jhonatan\_LAB12 (main) \$ █

2. Important considerations for this lab. If the code is run in a cmd or powershell application, use the commented alternative written in the source code. I added a lot of redundancies to handle the arguments of the functions. Instead of creating a write and append function, I merge them into one since I noticed that both functions were almost or even the same, the action parameter determines what will be performed. The same thing I did with the first .py file with loop and list. Lab12\_3.py commanded to create a function to read and display the content of a file, something that already was done in lab12\_1.py, and so that was the one I imported in the file module.