



READ/WRITE FILE AND THREAD

Ms.Thananya Supriyasilp
Team SDI

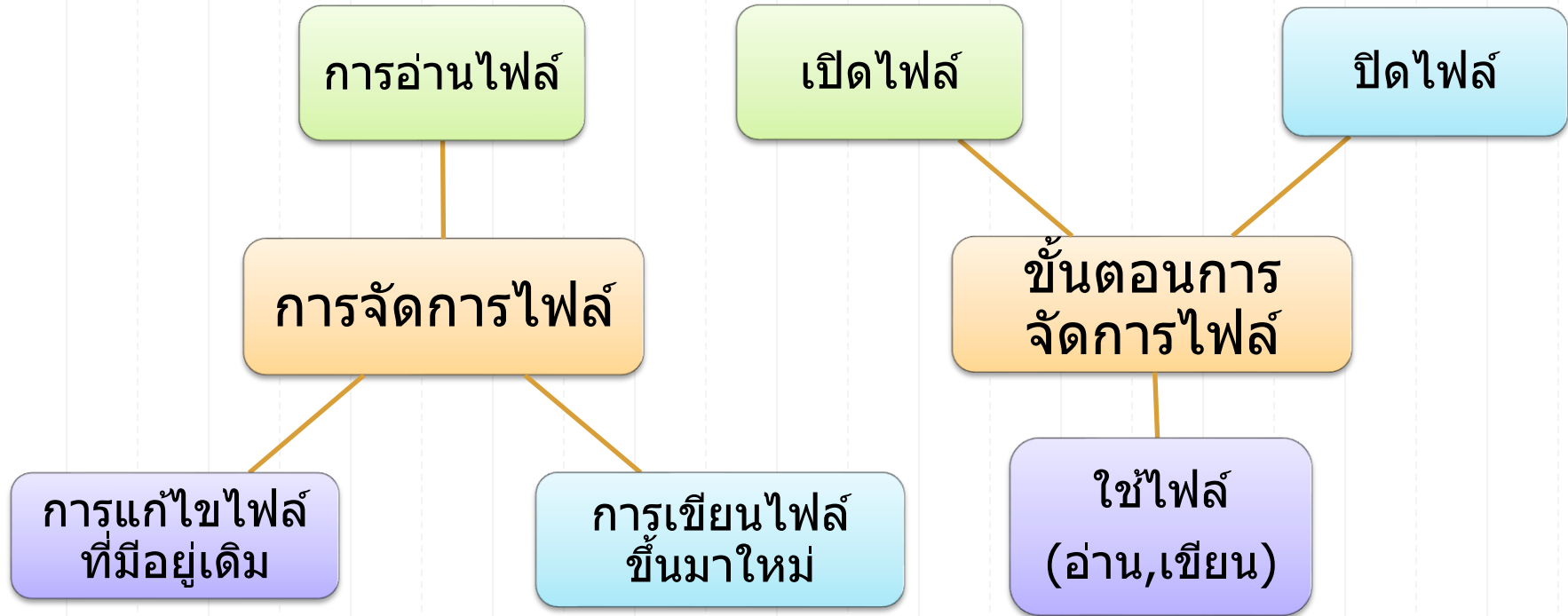
Content

- ◎ วิธีการและขั้นตอนการจัดการไฟล์
- ◎ การอ่านไฟล์ Yaml
- ◎ การเปิดไฟล์
- ◎ การประยุกต์ใช้ Yaml
- ◎ การอ่านละเขียนไฟล์
- ◎ การประมวลผลแบบขนาน
- ◎ การสร้างไฟล์ Yaml
- ◎ Threading Function



READ/WRITE FILE 1

วิธีการและขั้นตอนการจัดการไฟล์



OPEN: การเปิดไฟล์

โดยที่ filename เป็นชื่อหรือ Path ของไฟล์ที่ต้องการเปิด

```
f = open(filename, mode)
```

ฟังก์ชัน open() สำหรับเปิดไฟล์เพื่อวัตถุประสงค์ที่ต้องการ โดยฟังก์ชันสามารถเปิดเพื่อทำงานได้กับทั้ง Text file และ Binary file

mode เป็นโหมดสำหรับเปิดไฟล์ เช่น เปิดไฟล์เพื่อการอ่านหรือการเขียน เป็นต้น

OPEN: การเปิดไฟล์ด้วย Method With

```
with open(filename, mode) as f:
```

Tab คำสั่ง

- ฟังก์ชัน `open()` สำหรับเปิดไฟล์เพื่อวัตถุประสงค์ที่ต้องการ โดยฟังก์ชันสามารถเปิดเพื่อทำงานได้กับทั้ง Text file และ Binary file
- `f` คือ ตัวแปรแทนไฟล์ที่ต้องการเปิด

ตัวแปร mode และการใช้งาน

| Mode | Description |
|------|--|
| r | เปิดไฟล์เพื่ออ่านข้อมูลจากไฟล์ |
| w | เปิดไฟล์เพื่อเขียนข้อมูลลงบนไฟล์ |
| a | เปิดไฟล์เพื่อเขียนข้อมูลต่อท้ายไฟล์เดิม |
| r+ | เปิดไฟล์เพื่ออ่านข้อมูล หากไม่มีไฟล์จะสร้างไฟล์ใหม่ |
| w+ | เปิดไฟล์เพื่ออ่านและเขียนข้อมูล หากไม่มีไฟล์จะสร้างไฟล์ใหม่ |
| a+ | เปิดไฟล์เพื่ออ่านและเขียนข้อมูลต่อท้าย หากไม่มีไฟล์จะสร้างไฟล์ใหม่ |

Write: การเขียนลงบนไฟล์

Coding without method "with"

```
f = open("testfile.txt", mode)
f.write("Hello World!\n")
f.write("How are you today?\n")
f.write("I'm fine thank you.\n")
f.write("+++++")
f.close()
```

Coding with method "with"

```
With open("testfile.txt", mode) as f:
    f.write("Hello World!\n")
    f.write("How are you today?\n")
    f.write("I'm fine thank you.\n")
    f.write("+++++")
```


Write: การเขียนลงบนไฟล์

Coding

With `open("testfile.txt", mode)` as `f`:

```
f.write("Hello World!\n")
```

```
f.write("How are you today?\n")
```

```
f.write("I'm fine thank you.\n")
```

```
f.write("+++++")
```

```
f.write("eiei")
```



ValueError: I/O operation on closed file

Write: การเขียนลงบนไฟล์



Result

```
Hello World!  
How are you today?  
I'm fine thank you.  
+++++
```

โดยปกติแล้วในการทำงานกับไฟล์สามารถมีข้อผิดพลาดเกิดขึ้นได้ และจะทำให้โปรแกรมหยุดการทำงานในทันที ส่งผลให้ File object ไม่ถูกปิดอย่างถูกต้อง เราสามารถใช้งานไฟล์กับคำสั่ง with เพื่อยืนยันว่า File object จะถูกปิดการใช้งานเสมอไม่ว่าจะเกิดข้อผิดพลาดหรือไม่ก็ตาม



Read: การอ่านไฟล์

Coding

```
With open("testfile.txt", mode) as f:  
    print f.read()
```

Result

```
Hello World!  
How are you today?  
I'm fine thank you.  
+++++
```

- คำสั่ง `.read()` จะทำการอ่านข้อมูลทั้งหมดภายในครั้งเดียว
- คำสั่ง `.readline()` จะอ่านข้อมูลที่ละบรรทัด
- คำสั่ง `.readlines()` จะได้ผลลัพธ์มาเป็น list ของ string แต่ละบรรทัด

Read file with loop

Coding

```
With open("testfile.txt", mode) as f:  
    for i in range(5):  
        print f.readline()
```

```
With open("testfile.txt", mode) as f:  
    for line in f.readlines():  
        print line
```

Result

```
Hello World!  
How are you today?  
I'm fine thank you.  
++++++
```

Exercise 1

- รับค่าตัวเลขแบบทศนิยม 5 จำนวน
- เงื่อนไข
 - หาค่าที่มากที่สุด (Max)
 - หาค่าน้อยที่สุด (Min)
 - หาค่าเฉลี่ย (Mean)
 - เก็บค่าทั้งสามลงในไฟล์ .txt
 - แล้วอ่านผลลัพธ์ที่ได้จากไฟล์ .txt

Output

Max = A

Min = B

Mean = C



Yaml 2

Yaml: รูปแบบของไฟล์ .yaml

key:

Tab - value1

Tab - value2

Key: value

- **PyYAML**

- Yaml เป็นไฟล์ที่ใช้สำหรับแลกเปลี่ยนข้อมูลระหว่างการเขียนโปรแกรม เป็นภาษาที่สั้นและเข้าใจง่าย โดยไม่มีเครื่องหมายสัญลักษณ์เข้ามาเกี่ยวข้อง แต่จะใช้การเว้นวรรคในการแยกข้อมูลและคุณสมบัติของ Dictionary (Key, Value)

● วิธีการโหลด Libraly

- ไปที่เมนู Setting > Project Interpreter > กด + แล้วดาวน์โหลด Libraly ชื่อ "PyYaml"

● วิธีการสร้างไฟล์ Yaml

- คลิกขวา Folder > New > File
- ตั้งชื่อไฟล์ XXX.yaml

Yaml: การใช้ `yaml.load()`

profile.yaml

```
bob:
  age: 22
  color:
    - red
    - blue

alice:
  age: 25
  color:
    - black
    - pink
```

Coding

```
import yaml
with open("profile.yaml") as f:
    print yaml.load(f)
```

Result

```
{'bob': {'color': ['blue', 'red'], 'age': 22},
'alice': {'color': ['black', 'pink'], 'age': 25}}
```

การประยุกต์ใช้ข้อมูลจากไฟล์ Yaml

Coding

```
with open("profile.yaml") as f:  
    profile = yaml.load(f)  
print profile["bob"]  
print profile["bob"]["age"]  
print profile["alice"]  
print profile["alice"]["color"]
```

Result

```
{'color': ['blue', 'red'], 'age': 22}  
22  
{'color': ['black', 'pink'], 'age': 25}  
['black', 'pink']
```

การประยุกต์ใช้ข้อมูลจากไฟล์ Yaml

Coding

```
with open("profile.yaml") as f:  
    profile = yaml.load(f)  
for name in profile.keys():  
    if name == "bob":  
        print profile[name]
```

Result

```
{'color': ['blue', 'red'], 'age': 22}
```

การประยุกต์ใช้ข้อมูลจากไฟล์ Yaml

login.yaml

```
email:
  hostname: mail.inet.co.th
  username: email_address
  password: email_password
vm:
  ip: vm_ip
  port: 22
  username: vm_username
  password: vm_password
```

Coding

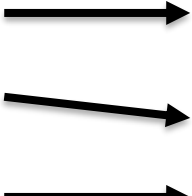
```
try:
    with open("login.yaml") as f:
        login = yaml.load(f)
except:
    print "ERROR: Can not open file yaml!"

hostname = login["email"]["hostname"]
email_username = login["email"]["username"]
email_password = login["email"]["password"]
```

การประยุกต์ใช้ข้อมูลจากไฟล์ Yaml

Coding

```
vm_ip = login["vm"]["ip"]  
vm_port = login["vm"]["port"]  
vm_username = login["vm"]["username"]  
vm_password = login["vm"]["password"]  
print login  
print hostname  
print vm_port
```



Result

```
{'email': {'username': 'email_address', 'password':  
'email_password', 'hostname': 'mail.inet.co.th'},  
'vm': {'username': 'vm_username', 'ip': 'vm_ip',  
'password': 'vm_password', 'port': 22}}  
mail.inet.co.th  
22
```

Exercise 2

- เก็บข้อมูลวันเกิด(วัน, เดือน, ปีเกิด) และรหัสพนักงาน ทั้งหมด 3 คน ตามรูปแบบของ Yaml ลงในไฟล์ .yaml
- เขียนโปรแกรม
 - แสดงค่าวันเกิดของคุณ
 - แสดงค่าเดือนเกิดของเพื่อน
 - แสดงค่ารหัสพนักงานของคุณ

Output

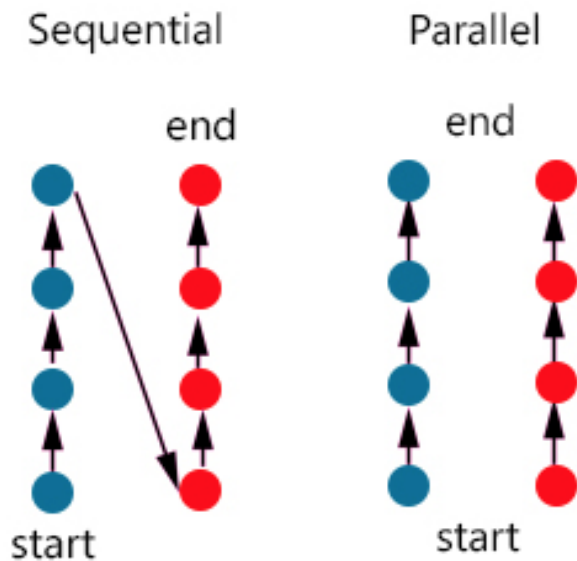
30

June

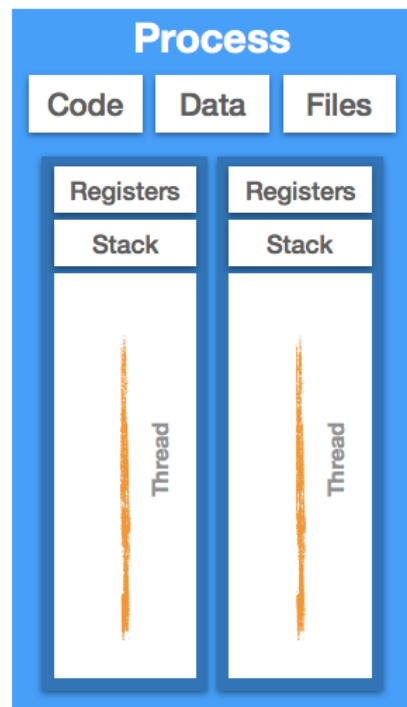
61306

THREAD

การประมวลผลแบบขนาน “



Single-threaded



Multithreaded

Thread: รูปแบบของ Thread Function

```
t = threading.Thread(target=function, args=(value,))
```

- t คือตัวแปรรับค่าฟังก์ชัน Threading
- ฟังก์ชัน threading.Thread() กำหนดการทำงาน และค่าตัวแปรที่ต้องการส่งไปยังฟังก์ชันนั้น ๆ
- คำสั่ง target=function คือการเรียกใช้ฟังก์ชันผ่านการเรียกชื่อ
- คำสั่ง args=(value,) คือ การส่งค่า value ไปยังฟังก์ชันที่เราเรียกใช้

Sequence Processing

Coding

```
import time
import threading
def cal_max(numbers, delay):
    time.sleep(delay)
    print "Max:", max(numbers)

def cal_min(numbers, delay):
    time.sleep(delay)
    print "Min:", min(numbers)

start_time = time.time()
```

- รับค่าได้มากกว่า 1 ค่า
- รับค่าเป็น list ได้

Sequence Processing

Coding

```
arr = [10, 7, 29, 62, 19]
```

```
delay1 = 3
```

```
delay2 = 5
```

```
cal_max(arr, delay1)
```

```
cal_min(arr, delay2)
```

```
print "Elapsed time:", time.time() - start_time
```

Result

Max: 62

Min: 7

Elapsed time: 8.02300000191

Parallel Processing: Thread

Coding

```
thread1 = threading.Thread(target=cal_max,  
args=(arr, delay1,))  
thread2 = threading.Thread(target=cal_min,  
args=(arr, delay2,))
```

```
thread1.start()  
thread2.start()
```



```
thread1.join()  
thread2.join()
```



```
print "Elapsed time:", time.time() - start_time
```

Result

Max: 62

Min: 7

Elapsed time: 5.01599979401

- คำสั่ง .start() เพื่อเริ่มการทำงานของ thread
 - Synchronize Thread
- คำสั่ง .join() คือการรอให้ thread สิ้นสุด
การทำงานเสร็จแล้วจึงเริ่มต้นทำงานโปรแกรม
ต่อไป

Sequence Processing

Coding

```
import requests
import threading
import time

def get_response(url):
    try:
        resp = requests.get(url)
        print url, "Status:" resp.status_code
    except:
        print url, "Fail"
    start_time = time.time()
```

- requests เป็น Library ที่มีไว้จัดการกับ HTTP
- requests.get() คือ คำสั่งที่เรียกดูข้อมูล เหมือนการอ่านหน้าเว็บนั้น ๆ
- .status_code คือ แสดงค่า HTTP status code

Sequence Processing

Coding

```
websites = ["http://www.google.com",  
            "http://www.facebook.com/admin",  
            "http://www.youtube.co.th/",  
            "http://www.twitter.com/",  
            "http://www.sanook.com/"]  
  
for web in websites:  
    get_response(web)  
  
print "Elapsed time:", time.time() - start_time
```

Result

<http://www.google.com> Status: 200
<http://www.facebook.com/admin> Status: 404
<http://www.youtube.co.th/> Status: 200
<http://www.twitter.com/> Status: 200
<http://www.sanook.com/> Status: 200
Elapsed time: 4.07899999619

Method "GET"

- Status Code: 200 OK
- Status Code: 404 Not Found

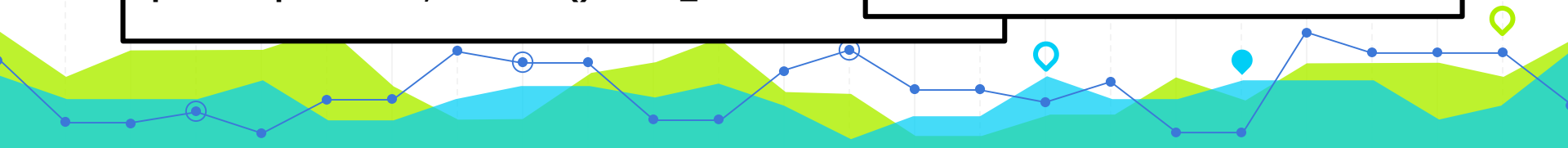
Parallel Processing: Thread

Coding

```
threads = []  
for website in websites:  
    t = threading.Thread(target=get_website, args=(website,))  
    threads.append(t)  
    t.start()  
  
for t in threads:  
    t.join()  
  
print "Elapsed time:", time.time() - start_time
```

Result

```
http://www.google.com Status: 200  
http://www.facebook.com/admin Status: 404  
http://www.youtube.co.th/ Status: 200  
http://www.twitter.com/ Status: 200  
http://www.sanook.com/ Status: 200  
Elapsed time: 1.80100011826
```



Exercise 3

- รับค่าทางหน้าจอจำนวน 4 ค่า
- คำนวณหา พื้นที่สี่เหลี่ยม และปริมาตรลูกบาศก์ โดยให้แต่ละฟังก์ชันมี Delay เท่ากับ 2 และ 3 วินาที
- เงื่อนไข
 - เขียนโปรแกรมแบบ Sequence Processing
 - เขียนโปรแกรมแบบ Parallel Processing
 - แสดงค่าเวลาที่ใช้ทั้งหมด

Output

พื้นที่สี่เหลี่ยม = ด้าน x ด้าน
ปริมาตรลูกบาศก์ = ด้าน x
ด้าน x ด้าน

Square = A

Cube = B

...

Elapsed time: C

THANK YOU!

Any questions?

contact

inet-sdi@inet.co.th

