

w14-Lec

# File Input/Output

for 204111

Kittipitch Kuptavanich

# Introduction to Unicode


แปลงจากตัวหนังสือ → เป็น Binary

- ในอดีต **ASCII** (**American Standard Code for Information Interchange**) เป็นมาตรฐานในการแสดงตัวอักษร โดยการใช้ตัวเลขระหว่าง 0 – 127 แทนตัวอักษรในภาษาอังกฤษ รวมถึงเครื่องหมายต่าง ๆ โดยการใช้ข้อมูลขนาด 1 byte (8 bit) เพื่อเก็บอักขระ 1 ตัว
- ปัญหา
  - Support เฉพาะตัวอักษรภาษาอังกฤษ
  - There was an 'e', but no 'é' or 'í' (accented characters)

# Introduction to Unicode [2]

- เนื่องจาก Unicode ใช้ตัวเลขในช่วง 0 – 127 คอมพิวเตอร์บางเครื่องจึงใช้ช่วง 128 – 255 สำหรับตัวอักษร accented
- มีหลายมาตรฐานในการแทนค่าอักษรเหล่านี้
  - 0 – 255 ไม่เพียงพอ
- ความพยายามในการสร้าง Unicode standard เริ่มในช่วงปี 1980s

# Introduction to Unicode [3]

- Unicode เริ่มจากการใช้ 16 bit ในการแทนข้อมูล ( $2^{16} = 65,536$ )  

- มีความตั้งใจที่จะแทนตัวอักษรได้จากทุกภาษา
  - 16 bit ไม่เพียงพอ
- UTF-8 เป็นมาตรฐาน UTF ที่ใช้มากที่สุด
  - Variable length with 8-bit code unit
  - ยังมี UTF-16 และ UTF-32
  - Compatible กับรหัส ascii ในช่วง 0 - 127

# UTF-8

- **UTF-8 uses the following rules:**
- ถ้ารหัสอยู่ในช่วง 0 – 127 ใช้เพียง 1 byte แทนตัวอักษรตามปกติ
- กรณีมากกว่า 127 UTF-8 จะแทนด้วย set ของ 2, 3, หรือ 4 bytes โดยแต่ละ byte มีค่า 128 - 255

เพิ่มคำสั่ง terminal `cd` `touch`  
`cat` ในการสร้าง เข้าถึงและ ดู  
ไฟล์

# Reading from Files

```

05 def read_file(filename, mode="rt"):
06     # rt = "read text"
07     with open(filename, mode, encoding='utf-8') as fin:
08         return fin.read()
09
10
11 print(read_file("./test.txt"))
12 print(read_file(r"./test.txt"))
13 print(read_file("./ทดสอบ.txt"))

```

เปิดไฟล์ในโหมด =  
แฉะหาว่าในคอม

อ่านไฟล์ที่ชื่อ open ก่อน แล้ว read

# File Operation

Character	Meaning
'r'	open for reading (default)
'w'	open for writing, truncating the file first
'x'	open for exclusive creation, <b>failing if the file already exists</b>
'a'	open for writing, <b>appending</b> to the end of the file if it exists
'b'	binary mode
't'	text mode (default)
'+'	open a disk file for updating (reading and writing)
'U'	<u>universal newlines</u> mode (deprecated)



# File Operation [2]

## Binary Files

- On Windows, **'b'** appended to the mode opens the file in binary mode, so there are also modes like
  - **'rb'**, **'wb'**, and **'r+b'**.
  - Python on Windows makes a distinction between text and binary files;
- On Unix, it doesn't hurt to append a **'b'** to the mode, so you can use it platform-independently for all binary files.

# Writing to Files

```
05 def write_file(filename, contents, mode="wt"):
06     # wt = "write text"
07     with open(filename, mode) as fout:
08         fout.write(contents)
09
10 write_file("./testout.txt", "hello")
```

# Deleting Files

```
05 import os
06
07 def delete_file(path):
08     if (os.path.exists(path)):
09         try:
10             os.remove(path)
11             return True → remove 7.
12         except FileNotFoundError:
13             return False
14     else:
15         return False
16
17
18 print(delete_file('./hello.txt'))
```

# Reading from Files [2]

```
05 def read_file(filename, mode="rt"):
06     # rt = "read text"
07     try:
08         with open(filename, mode, encoding='utf-8') as fin:
09             return fin.read()
10     except FileNotFoundError:
11         return None
12
13
14 content = read_file("./scores.txt")
15
16 if content:
17     ...
```

# I/O Redirection - Recap

## กรณีจำนวน Input ไม่ตายตัว

### • Example 1: กรณีทราบจำนวน

- หากจำนวน Input เป็นแบบ Dynamic คือไม่ตายตัว แต่ทราบจำนวน
  - เช่น มีการระบุว่า Input ตัวแรกจะเป็น จำนวนนักศึกษาทั้งหมดที่ต้องการหาคะแนนเฉลี่ย

```

05 def find_mean():
06     total = int(input())
07     sum_ = 0
08     ...
09
10     for i in range(total):
11         current_score = float(input())
12         sum_ = sum_ + current_score
13         ...

```

scores.txt

5  
22.3  
26.5  
33.6  
29.2  
13.1

# Modifying Code from I/O Redirection

## กรณีจำนวน Input ไม่ตายตัว

### • Example 1: กรณีทราบจำนวน

- หากจำนวน Input เป็นแบบ ...
- เช่น มีการระบุว่า Input ต้องการหาคะแนนเฉลี่ย

หาจำนวนบรรทัดได้จาก len()

```


05 def find_mean(content):
06     lines = content.splitlines()
07     total = int(lines[0])
08     sum_ = 0
09     # lines: ['5', '22.3', '26.5', ...]
10     for i in range(total):
11         current_score = float(lines[i+1])
12         sum_ = sum_ + current_score
13         ...

```

26.5  
33.6  
29.2  
13.1

# More Examples

```
05 def smiley_print(filename):
06     with open(filename, "rt") as f:
07         for line in f:
08             print(":)", line, end='')
09
10 smiley_print("sample.txt")
11
12 def copy_with_smilies(source, destination):
13     with open(source, "rt") as infile:
14         with open(destination, "wt") as outfile:
15             for line in infile:
16                 outfile.write(":) " + line)
17
18 copy_with_smilies("sample.txt", "smilies.txt")
19
20
```



# More Examples [2]

```
05 import os
06
07 for file in os.listdir("."):
08     # "." (dot) means the current directory
09     if ((len(file) > 4) and (file[-4:] == ".txt")):
10         print("TEXT FILE: ", file)
11         print("I'm going to SMILEY it")
12         smiley_print(file)
13
```




# More Examples [3]

```
05 # writing to the same file (using a buffer)
06
07 def smiley_add_buffered(filename):
08     buff = []
09     with open(filename, "rt") as f:
10         for line in f:
11             buff += [":)" + line]
12     destination = filename
13     with open(destination, "wt") as outfile:
14         for line in buff:
15             outfile.write(line)
16
17
18 smiley_add_buffered("./sample.txt")
```

เขียนทับไฟล์เก่า

# More Examples [4]

```
05 # writing to the same file (using a temp file)
06 import os
07
08 def smiley_add_temp(source):
09     ext = source.split('.')[-1]
10     temp_name = ','.join(source.split('.')[:-1]) + '_TEMP' + ext
11     
12     with open(source, "rt") as infile:
13         with open(temp_name, "wt") as outfile:
14             for line in infile:
15                 outfile.write(":) " + line)
16
17     os.rename(temp_name, source)
18
19 smiley_add_temp('./sample.txt')
21
```

# Reading from URL

```
05 from urllib.request import urlopen
06
07
08 def read_web_page(url):
09     assert url.startswith("https://")
10     with urlopen(url) as res:
11         return res.read()
12
13
14 print(read_web_page("https://www.python.org")[:10])
```

# Loading JSON from URL

```
15 # Seoul - Korea
16 url = "https://api.open-
    meteo.com/v1/forecast?latitude=37.5139&longitude=12
    6.9828&hourly=temperature_2m"
```

```
{
  "latitude":52.52,
  "longitude":13.419998,
  "elevation":44.8125,
  "hourly": {
    "time":["2022-07-01T00:00","2022-07-01T01:00", ...]
    "temperature_2m":[13.7,13.3,12.8,12.3,11.8, ...]
  }
}
```



Temperature in metric system

# Loading JSON from URL [2]

```
05 import json
06 from urllib.request import urlopen
07
08 ..
16 url = "https://api.open-
    meteo.com/v1/forecast?latitude=37.5139&longitude=126.9828
    &hourly=temperature_2m"
17
18 # store the response of URL
19 response = read_web_page(url)
20
21 # storing the JSON response
22 data_json = json.loads(response)
23
24 print('time =', data_json['hourly']['time'][0], end='')
25 print('\ttemp =',
26       data_json['hourly']['temperature_2m'][0], end='')
    time = 2022-09-27T00:00 temp = 16.5 C
```

# Practice 1

- ให้สร้างฟังก์ชันสำหรับนำเข้าข้อมูล Matrix จัตุรัส ที่มีจำนวน row เป็นเลขคี่ตั้งแต่ 3 ขึ้นไป โดยที่ค่าข้อมูลนำเข้าจากการอ่านไฟล์จะเป็นตัวเลขจำนวนเต็ม 0 หรือ 1 เท่านั้น (อยู่ในลักษณะ เลขฐาน 2) หลังจากนั้นให้แปลงตัวเลขที่อยู่ในแต่ละทิตซ์ของ Matrix ทั้ง 4 ทิตซ์ จากเลขฐาน 2 เป็นเลขฐาน 10 และ return ค่า list ของเลขฐาน 10 ที่แปลงได้ทั้ง 4 ค่า

1	1	0
0	1	0
0	0	1

$$110_2 = 6_{10}$$

$$010_2 = 2_{10}$$

$$111_2 = 7_{10}$$

$$010_2 = 2_{10}$$

returns [6, 2, 7, 2]

0	0	1	1	0
1	0	1	1	0
1	1	0	1	0
0	0	0	0	0
1	0	1	0	1

$$11001_2 = 25_{10}$$

$$11010_2 = 26_{10}$$

$$00001_2 = 1_{10}$$

$$10010_2 = 18_{10}$$

returns [25, 26, 1, 18]

# References

- <http://farmdev.com/talks/unicode/>
- <http://www.kosbie.net/cmu/fall-14/15-112/>
- <http://www.kosbie.net/cmu/spring-13/15-112/handouts/fileWebIO.py>
- <http://www.kosbie.net/cmu/spring-15/15-112/notes/notes-functions-redux-and-web-and-file-io.html>
- <http://www.cs.cmu.edu/~nschneid/pythonathan/>
- <https://docs.python.org/3/howto/unicode.html>