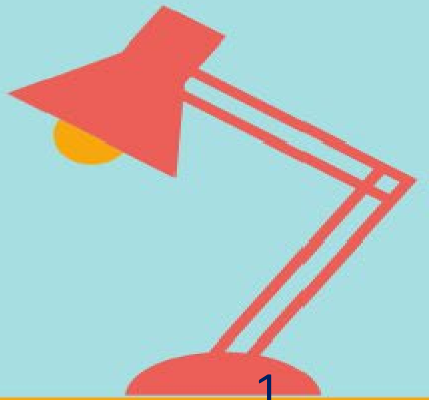
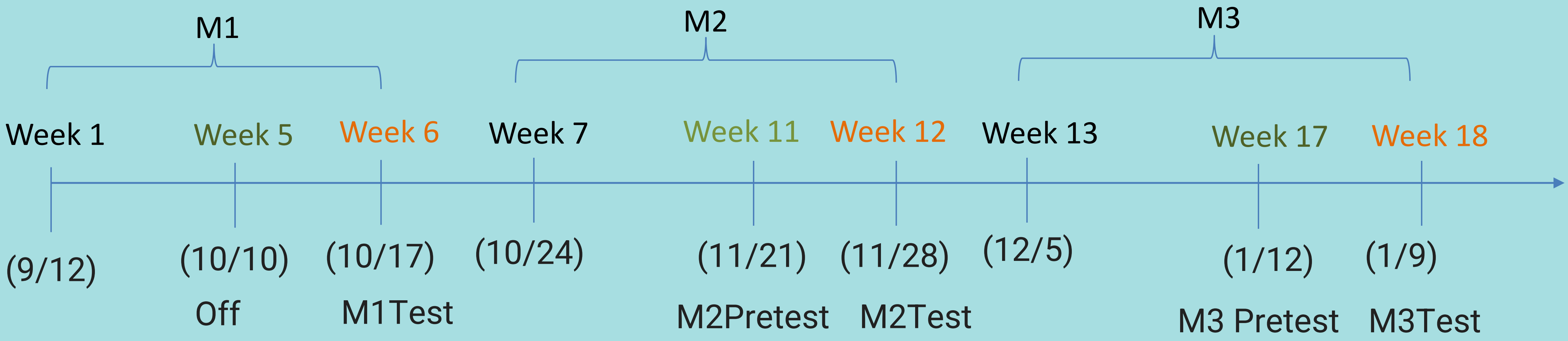
Fundamental Programming Course
Week 17

Huseh-Ting Chu@Asia University, 2022



Schedule



M1

M2

M3

Week 1 (9/12)

Week 5 (10/10) Off

Week 6 (10/17) M1Test

Week 7 (10/24)

Week 11 (11/21) M2Pretest

Week 12 (11/28) M2Test

Week 13 (12/5)

Week 17 (1/12) M3 Pretest

Week 18 (1/9) M3Test

Syllabus

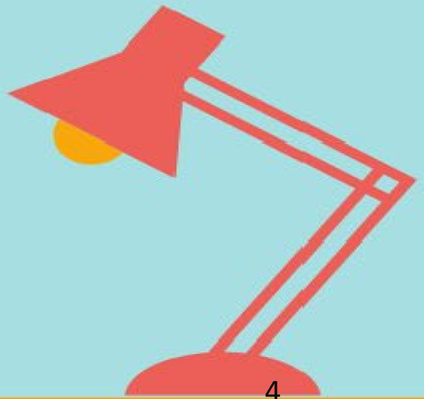
- W1-Python Introduction and Programming Tools
- W2-Variables and Operations
- W3-Loop and formatted output
- W4-Condition and Containers
- W5-String and built-in functions
- W6-M1 test

- W07-Dictionary Container
- W08-File I/O
- W09-Function
- W10-Advanced flow control
- W11-Advanced operations and generators
- W12-M2 test

- W13-Advanced functions (definitions and calls, Recursion)
- W14-Class fundamentals (classes, objects, properties, constructors, methods)
- W15-Advanced Classes (Static methods, class Methods and class decorators)
- W16-Modules and Packages
- W17-Advanced programming(Argparse)
- W18-M3 test

Week 17 topics

- Week17-Advanced programming
  - Topic 1-urllib
  - Topic 2-SQLite
  - Topic 3-argparse (new since PEP 389 , 2009/09)
  - Topic 4-getopt (old)
  - Topic 5-Review of built-in functions and libraries



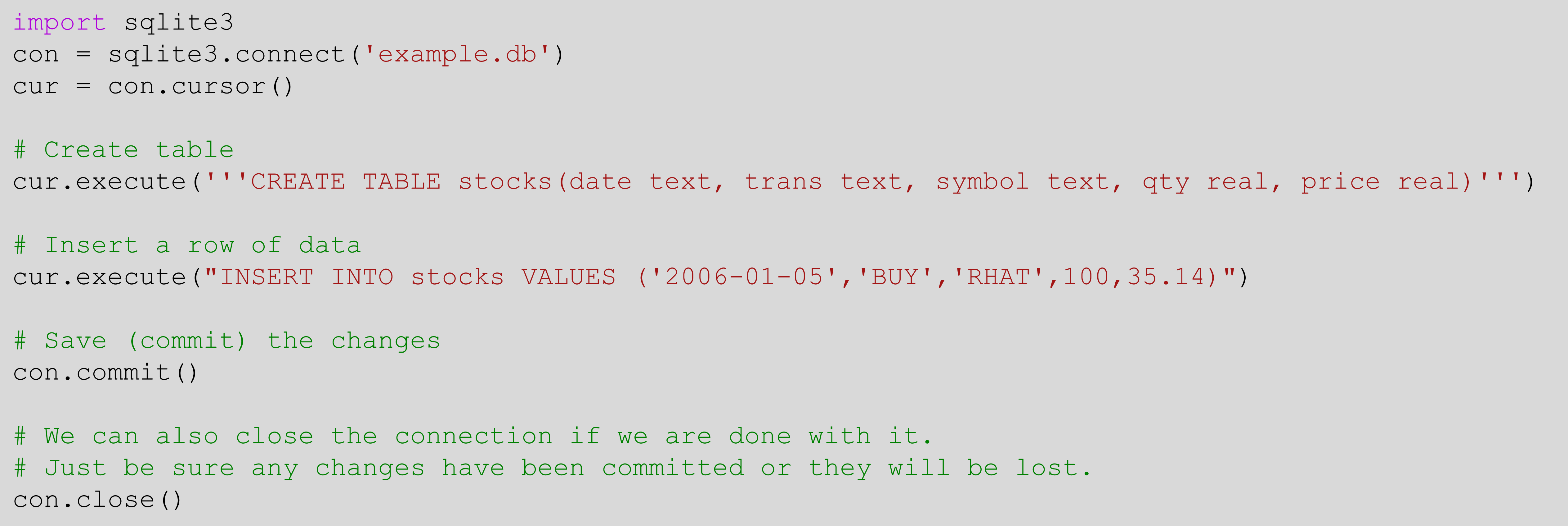
Topic 1- urllib函數庫

urllib.request is a Python module for fetching data from URLs (Uniform Resource Locators). It provides a very simple interface to accept many different protocols, urlopen function. It also provides a more complex interface for handling some common situations, such as: basic authentication, cookies, proxies, etc., which can be operated by handler or opener objects.

```python
import urllib.request
with urllib.request.urlopen('http://www.asia.edu.tw/') as response:
   html = response.read()
```

Topic 2-SQLite DB-API 2.0 interface

- Connection object
- Cursor object



```
import sqlite3
con = sqlite3.connect('example.db')
cur = con.cursor()

# Create table
cur.execute('''CREATE TABLE stocks(date text, trans text, symbol text, qty real, price real)''')

# Insert a row of data
cur.execute("INSERT INTO stocks VALUES ('2006-01-05','BUY','RHAT',100,35.14)")

# Save (commit) the changes
con.commit()

# We can also close the connection if we are done with it.
# Just be sure any changes have been committed or they will be lost.
con.close()
```

SQL (Structured Query Language)

- DDL: data definition language

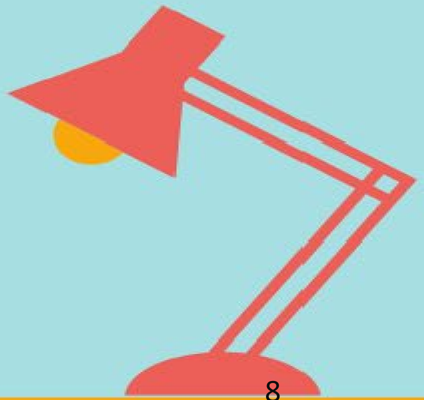
```
CREATE TABLE Books
(Id INT PRIMARY KEY IDENTITY(1,1),
Name VARCHAR (50) NOT NULL,
Price INT)
```

- DML:

```
INSERT into students values(1,'ashish','java’);
INSERT into students values(2,’rahul’,’C++’);
SELECT * from students;
```

Topic 3- argparse

- argparse- **New Command Line Parsing Module**
- getopt - Parser for command line options



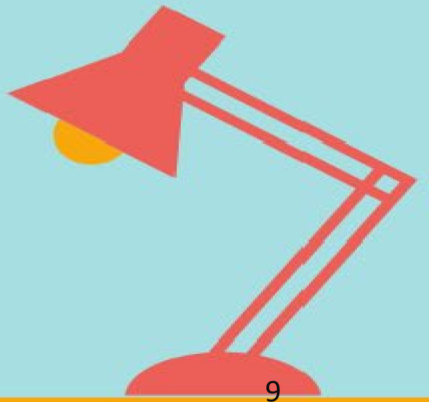
Topic 3- argparse

```
#prog.py
import argparse
parser = argparse.ArgumentParser()
parser.parse_args()
```

```
>python3 prog.py
>python3 prog.py --help
usage: prog.py [-h]

options:
  -h, --help  show this help message and exit

>python3 prog.py --verbose
usage: prog.py [-h]
prog.py: error: unrecognized arguments: --verbose
>python3 prog.py foo
usage: prog.py [-h]
prog.py: error: unrecognized arguments: foo
```



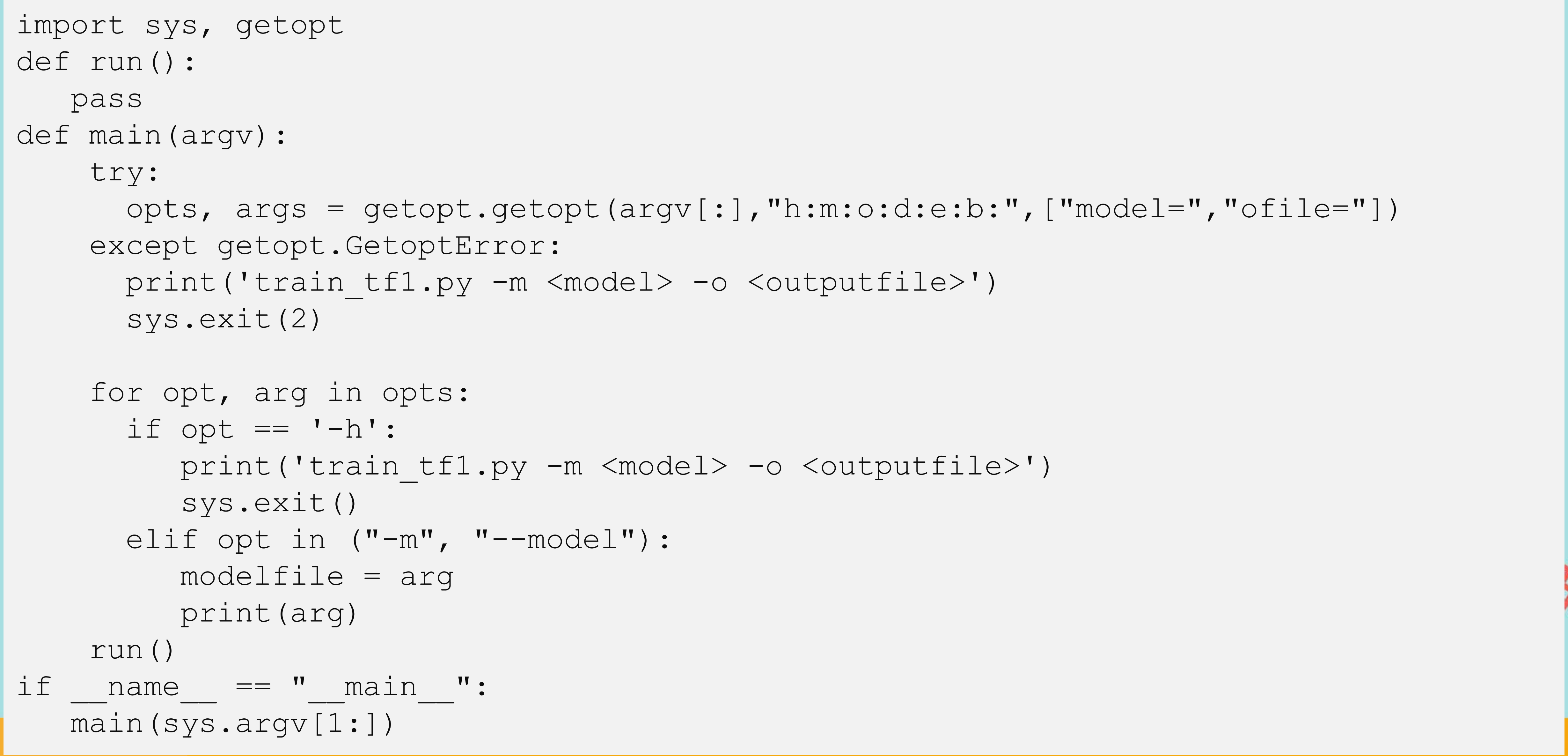
https://docs.python.org/zh-tw/3/howto/argparse.html

```
import argparse
def TrainModel(args):
    num_epochs = args.epochs
    batches = args.batches
    imgdir = args.imagedir
    modelfile = args.inputmodel
    outputfile = args.outputmodel
    model_select = args.model

if __name__ == '__main__':
    parser = argparse.ArgumentParser()
    parser.add_argument("-i", "--inputmodel", default=None, help = "input model")
    parser.add_argument("-o", "--outputmodel", default=“1.pth", help = "out model")
    parser.add_argument("-y", "--imagedir", default="images2", help = "train images")
    parser.add_argument("-e", "--epochs", default=1,type=int,  help = "train images")
    parser.add_argument("-b", "--batches", default=6, type=int, help = "train images")
    parser.add_argument("-c", "--cuda", default=0, type=int, help = "cuda device")
    parser.add_argument("-m", "--model", default=3, type=int, help = "select model")
    args = parser.parse_args()
    TrainModel(args)
```

```
python train2a.py -i fl2_v2s_e5.pth -o fl2_v2s_e10.pth -e 5 -c 0 -b 36 -m 11
```

Topic 4- getopt

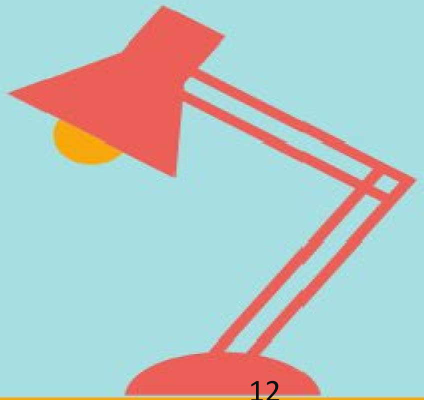


```
import sys, getopt
def run():
   pass
def main(argv):
    try:
      opts, args = getopt.getopt(argv[:],"h:m:o:d:e:b:",["model=","ofile="])
    except getopt.GetoptError:
      print('train_tf1.py -m <model> -o <outputfile>')
      sys.exit(2)

    for opt, arg in opts:
      if opt == '-h':
         print('train_tf1.py -m <model> -o <outputfile>')
         sys.exit()
      elif opt in ("-m", "--model"):
         modelfile = arg
         print(arg)
    run()
if __name__ == "__main__":
   main(sys.argv[1:])
```

Topic 5-內建函數和函數庫的複習

- sorting
- The day of the week and the number of days on the first day of each month



Thanks!

Q&A