Python Real Application

Getting Started

About Us

國立臺北科技大學資工系

Instructor: 洪子軒

TA: 林冠璋

Why Python?

Installation

Easy to install

Module

pip install moduleName

Execution

```
python3 filename.py
```

Syntax

Easy and Beautiful

Version

Python3.6

About Python

Dynamic Language: 寫法靈活

Modularity: 很多Module提供使用

Our Goal

1. Basic Python Skill
2. Basic OOP Concept
3. Basic Database Concept
4. Basic Tkinter Concept
5. Basic Web Concept

D1-Outline

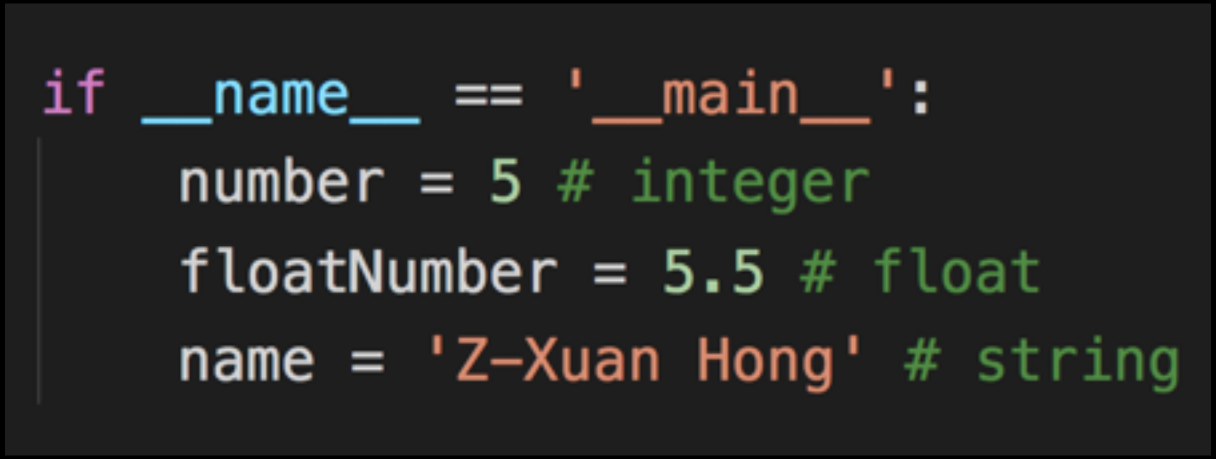
1. Variable & Data Type
2. Operator
3. Data Structure
4. Condition & Loop
5. Function

Let’s Build A HelloWorld APP

Python I / O

print( )
input( )

Variable & Data Type

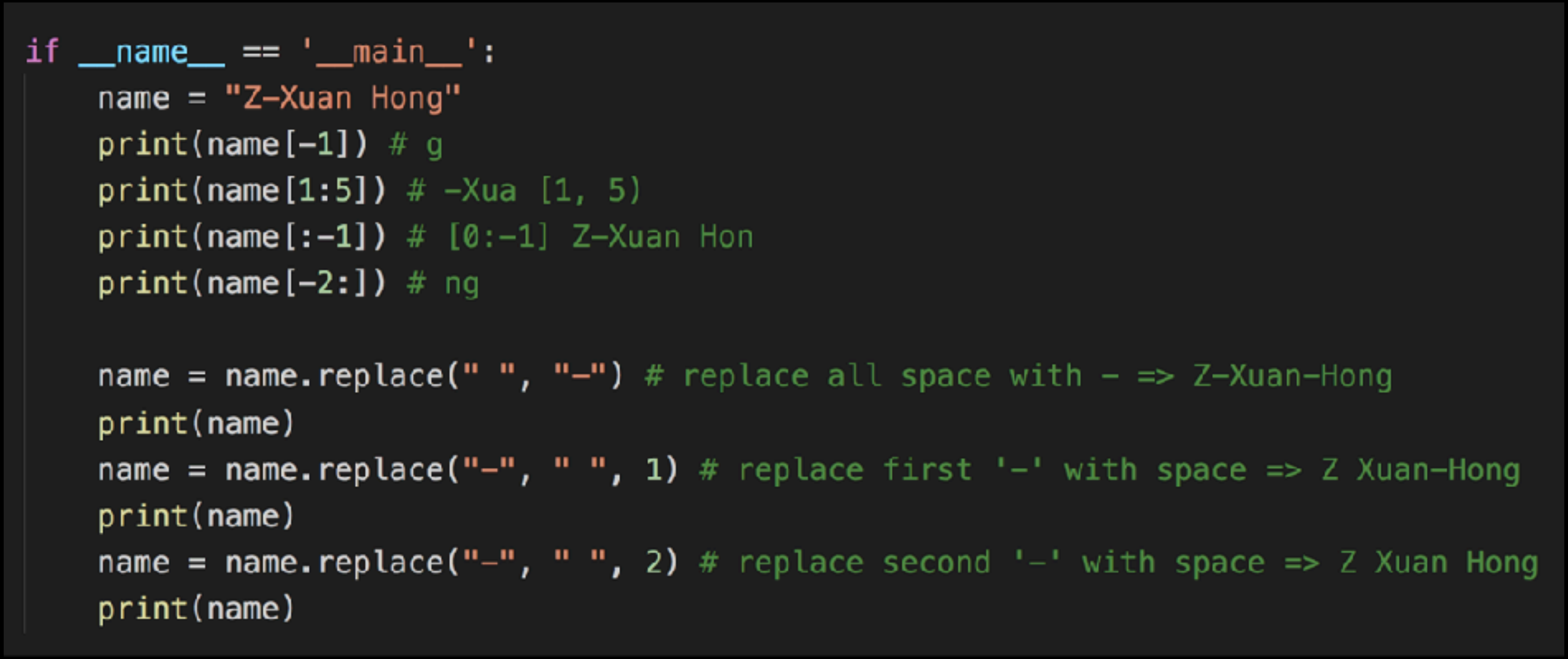


```python
if __name__ == '__main__':
    number = 5 # integer
    floatNumber = 5.5 # float
    name = 'Z-Xuan Hong' # string
```

Indexing

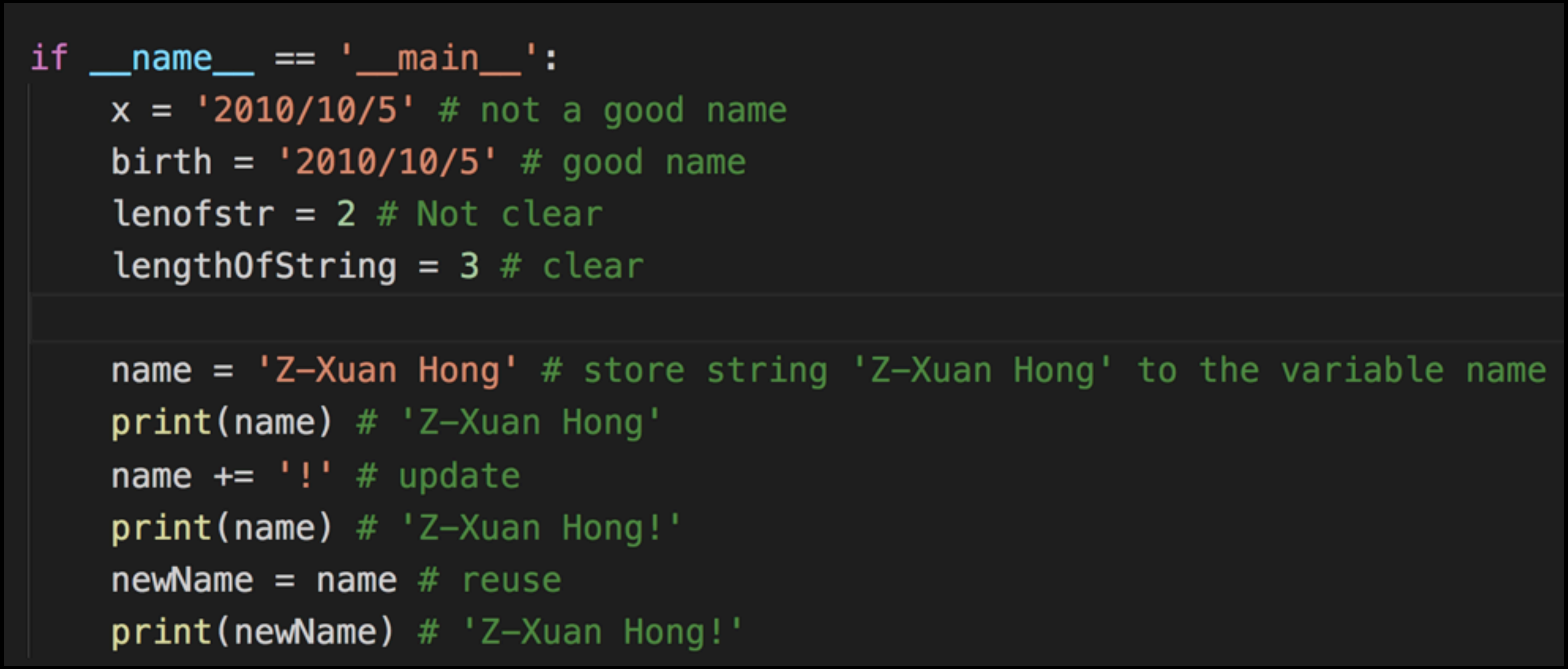
用來索引節料結構

Variable & Data Type



```
if __name__ == '__main__':
    name = "Z-Xuan Hong"
    print(name[-1]) # g
    print(name[1:5]) # -Xua [1, 5)
    print(name[:-1]) # [0:-1] Z-Xuan Hon
    print(name[-2:]) # ng

    name = name.replace(" ", "-") # replace all space with - => Z-Xuan-Hong
    print(name)
    name = name.replace("-", " ", 1) # replace first '-' with space => Z Xuan-Hong
    print(name)
    name = name.replace("-", " ", 2) # replace second '-' with space => Z Xuan Hong
    print(name)
```

Variable Name

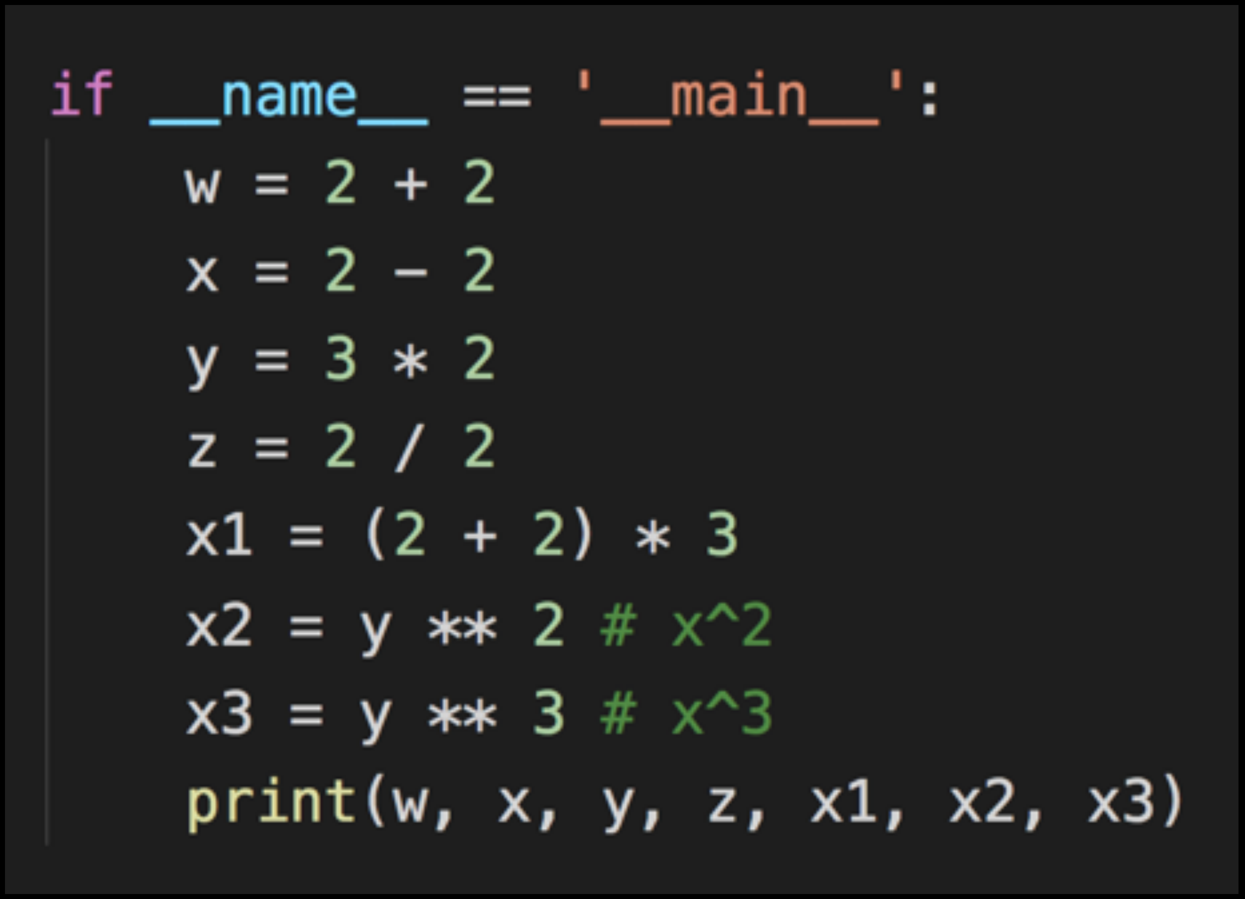


```
if __name__ == '__main__':
    x = '2010/10/5' # not a good name
    birth = '2010/10/5' # good name
    lenofstr = 2 # Not clear
    lengthOfString = 3 # clear

    name = 'Z-Xuan Hong' # store string 'Z-Xuan Hong' to the variable name
    print(name) # 'Z-Xuan Hong'
    name += '!' # update
    print(name) # 'Z-Xuan Hong!'
    newName = name # reuse
    print(newName) # 'Z-Xuan Hong!'
```

Simple Rule

變數的開頭小寫、不縮寫、之後的字首都要大寫

Operator



```python
if __name__ == '__main__':
    w = 2 + 2
    x = 2 - 2
    y = 3 * 2
    z = 2 / 2
    x1 = (2 + 2) * 3
    x2 = y ** 2 # x^2
    x3 = y ** 3 # x^3
    print(w, x, y, z, x1, x2, x3)
```

Operator Rule

括號裡面先運算、基本四則運算規則

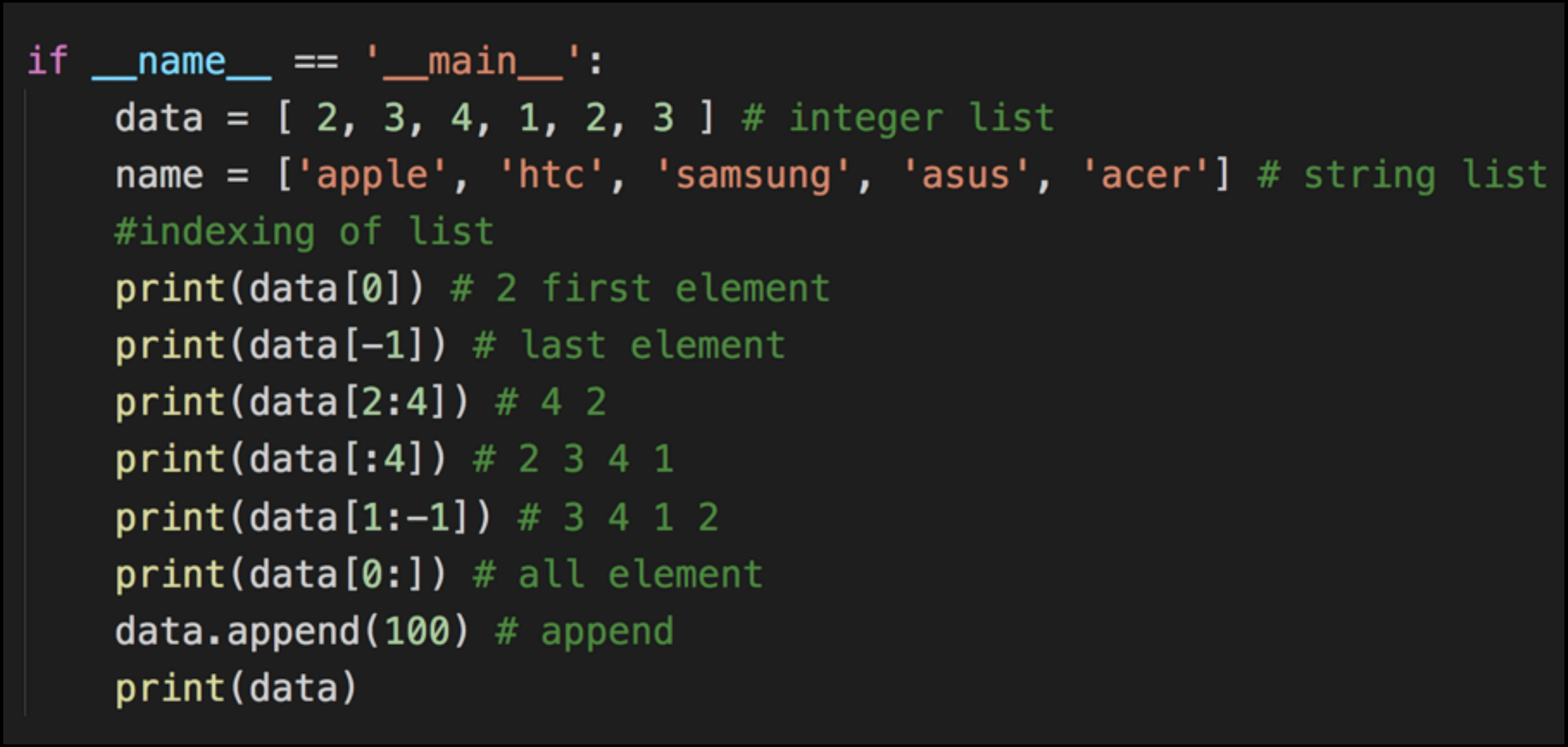
Data Structure

儲存資料的容器，每個資料結構有自己的特性，
新增，修改，刪除，取值

Data Structure

List & Dict

List [ ]

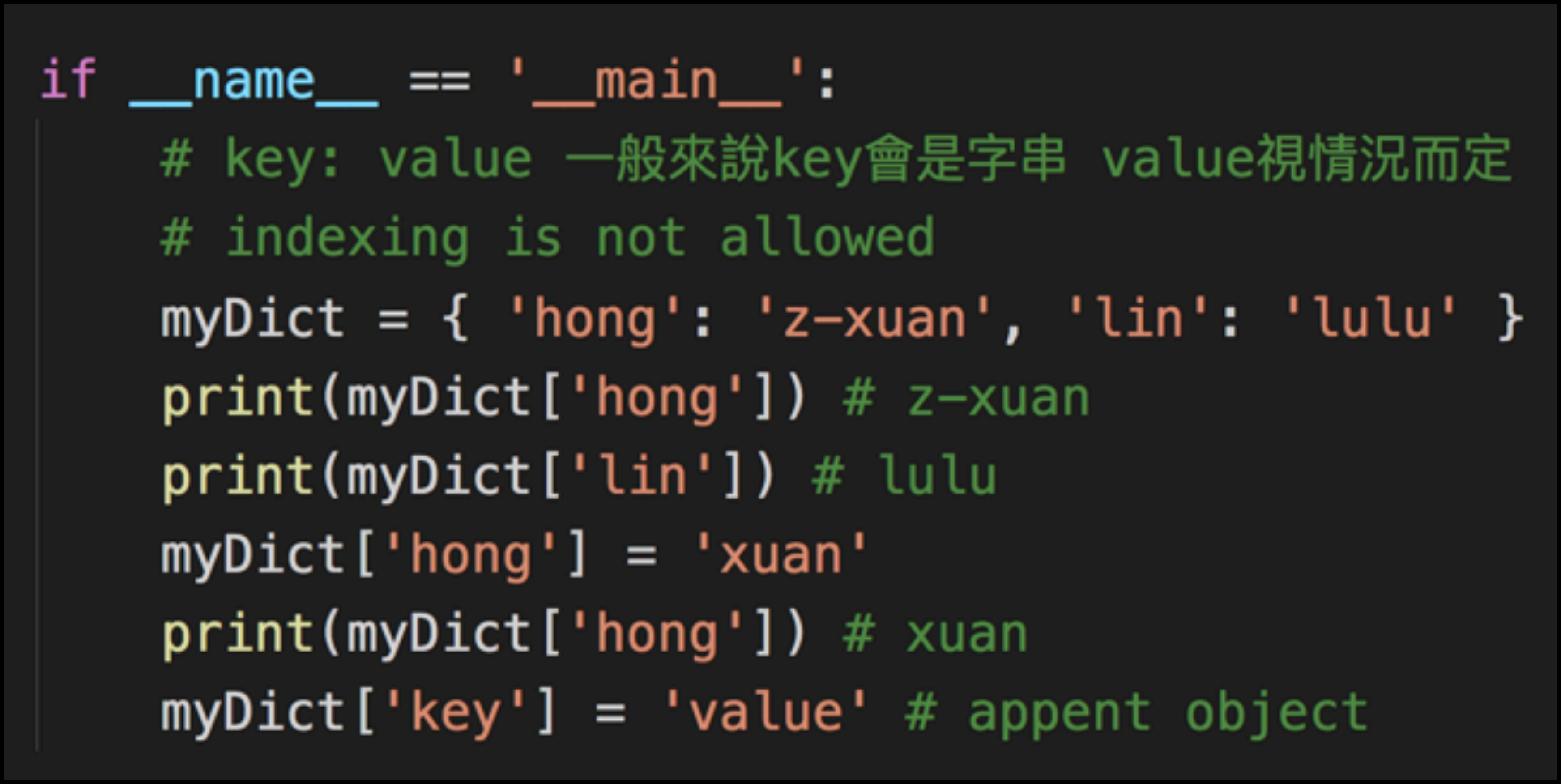


```
if __name__ == '__main__':
    data = [ 2, 3, 4, 1, 2, 3 ] # integer list
    name = ['apple', 'htc', 'samsung', 'asus', 'acer'] # string list
    #indexing of list
    print(data[0]) # 2 first element
    print(data[-1]) # last element
    print(data[2:4]) # 4 2
    print(data[:4]) # 2 3 4 1
    print(data[1:-1]) # 3 4 1 2
    print(data[0:]) # all element
    data.append(100) # append
    print(data)
```

Simple Quiz

建立一個List 裡面儲存Apple,、Banana、
Peach字串，用index全部print出來。

Dict { Key: Value }

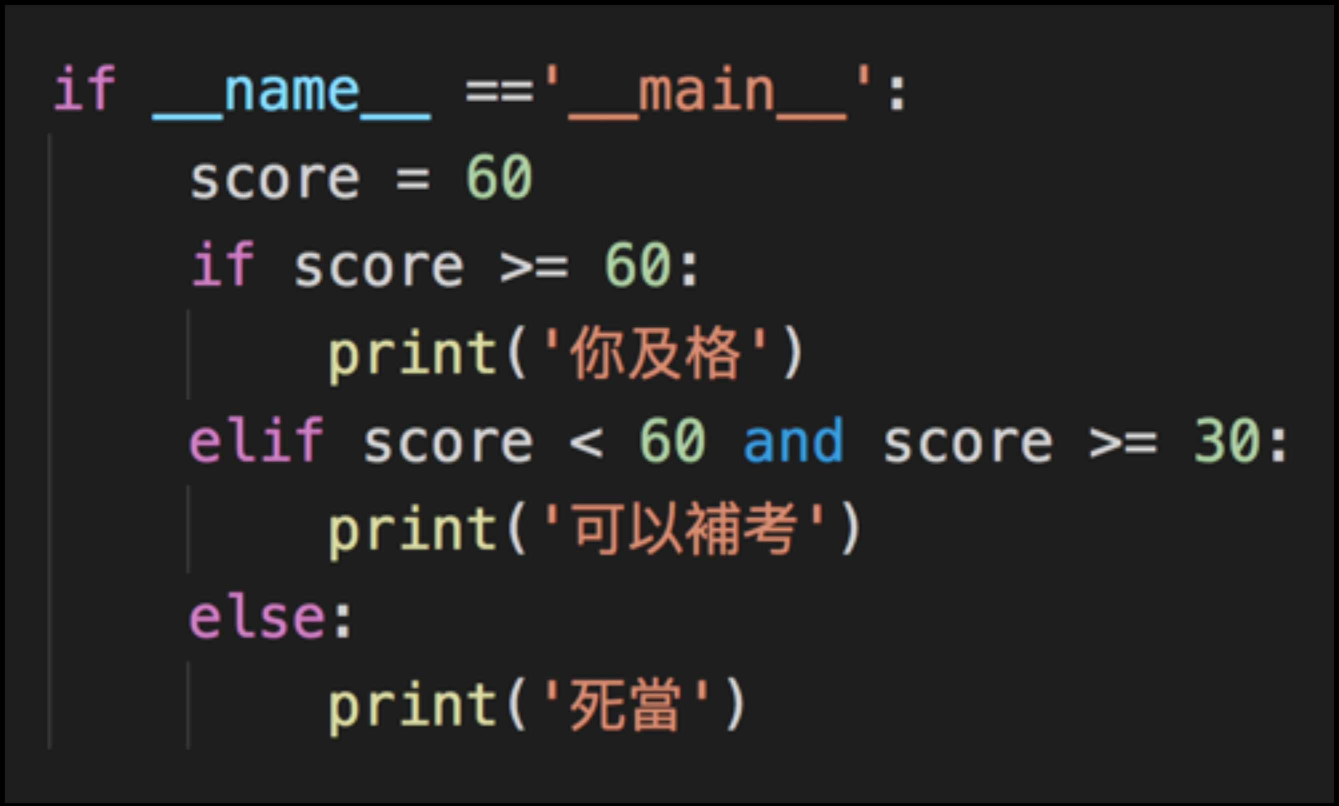


```python
if __name__ == '__main__':
    # key: value 一般來說key會是字串 value視情況而定
    # indexing is not allowed
    myDict = { 'hong': 'z-xuan', 'lin': 'lulu' }
    print(myDict['hong']) # z-xuan
    print(myDict['lin']) # lulu
    myDict['hong'] = 'xuan'
    print(myDict['hong']) # xuan
    myDict['key'] = 'value' # appent object
```

Simple Quiz

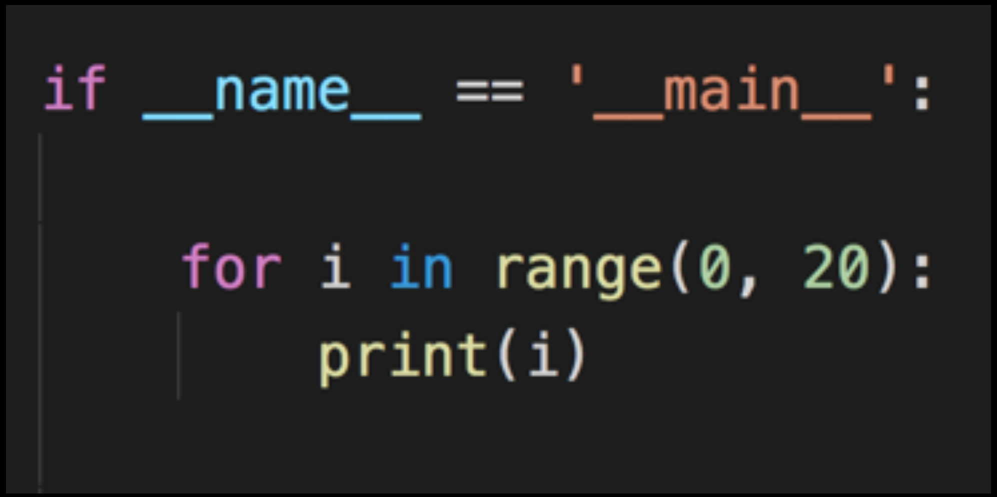
建立一個Dict裡面儲存，key跟value一樣，並且print出來

Condition & Loop



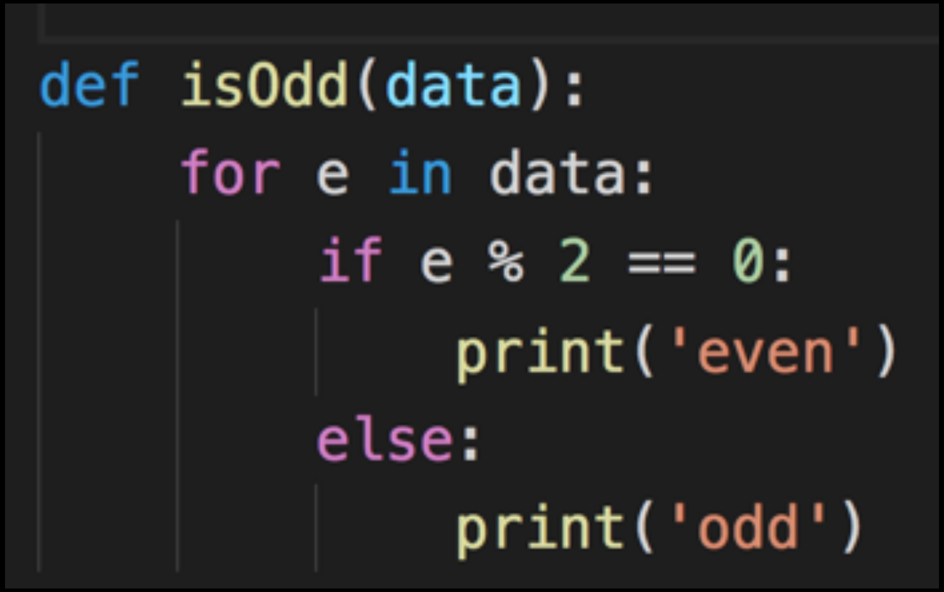
```python
if __name__ =='__main__':
    score = 60
    if score >= 60:
        print('你及格')
    elif score < 60 and score >= 30:
        print('可以補考')
    else:
        print('死當')
```

Condition & Loop



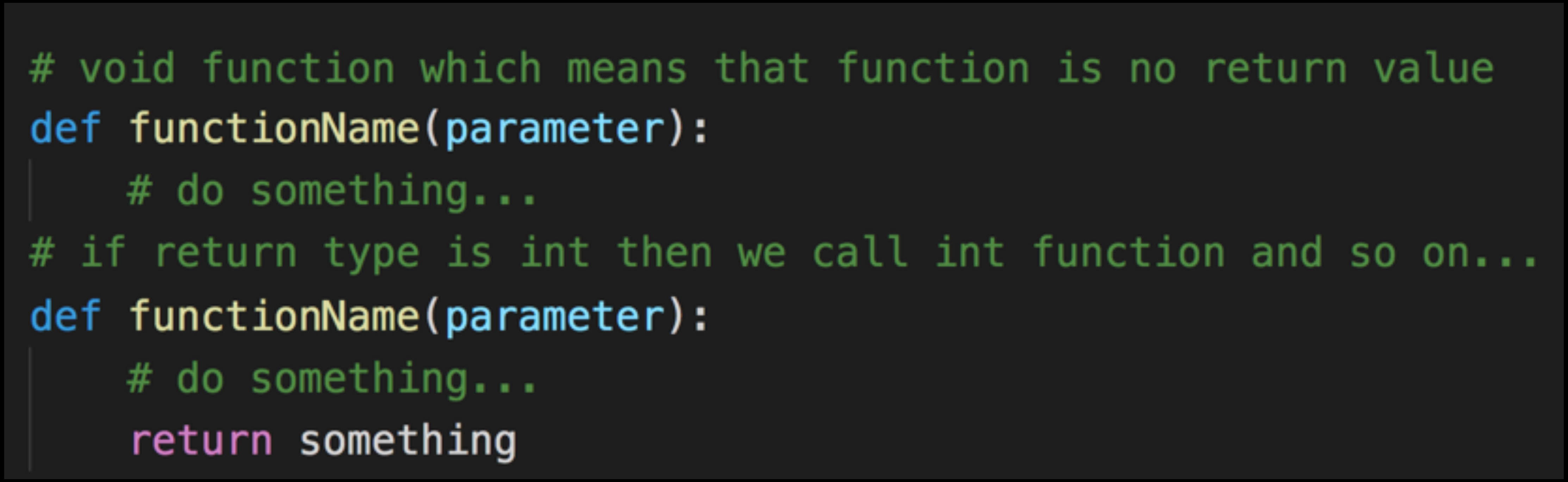
```
if __name__ == '__main__':

    for i in range(0, 20):
        print(i)
```

Condition & Loop



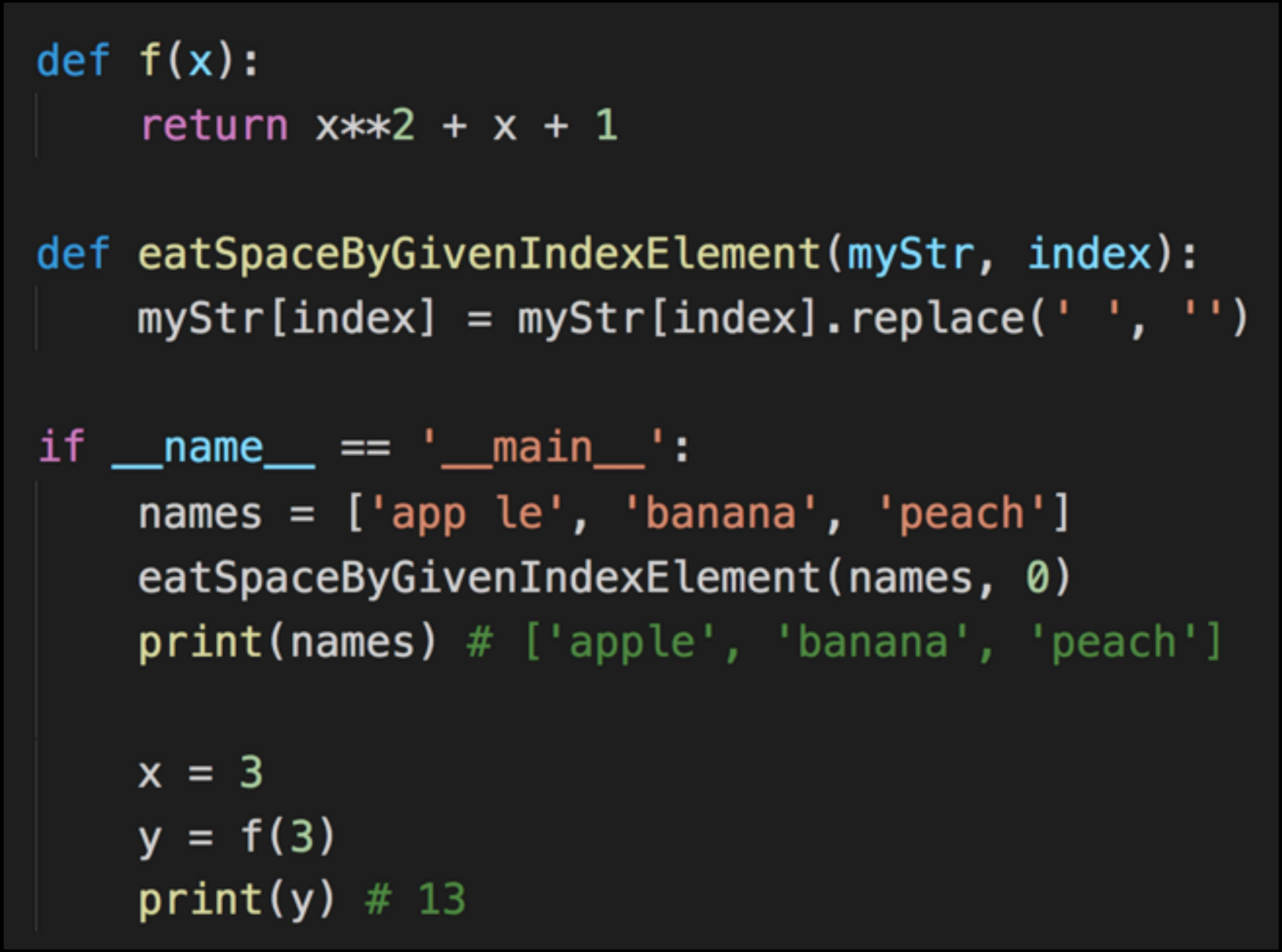
```python
def isOdd(data):
    for e in data:
        if e % 2 == 0:
            print('even')
        else:
            print('odd')
```

Function



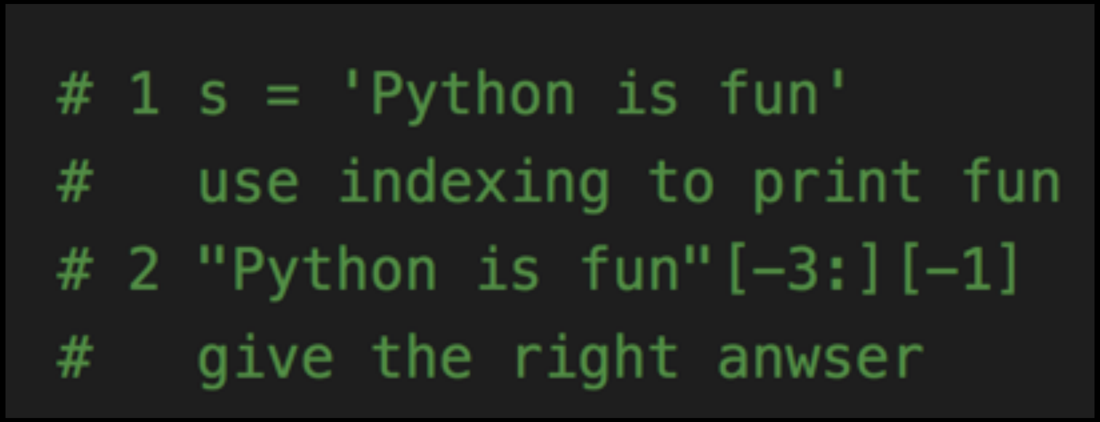
```python
# void function which means that function is no return value
def functionName(parameter):
    # do something...
# if return type is int then we call int function and so on...
def functionName(parameter):
    # do something...
    return something
```

Function



```python
def f(x):
    return x**2 + x + 1

def eatSpaceByGivenIndexElement(myStr, index):
    myStr[index] = myStr[index].replace(' ', '')

if __name__ == '__main__':
    names = ['app le', 'banana', 'peach']
    eatSpaceByGivenIndexElement(names, 0)
    print(names) # ['apple', 'banana', 'peach']

    x = 3
    y = f(3)
    print(y) # 13
```

Simple Quiz



```
# 1 s = 'Python is fun'
#   use indexing to print fun
# 2 "Python is fun"[-3:][-1]
#   give the right anwser
```

Simple Quiz

寫個函數可以把C轉F

hint: F = C*9/5 + 32

Simple Quiz

限制剛剛寫的函數，C不能<=-273.15。

Simple Quiz

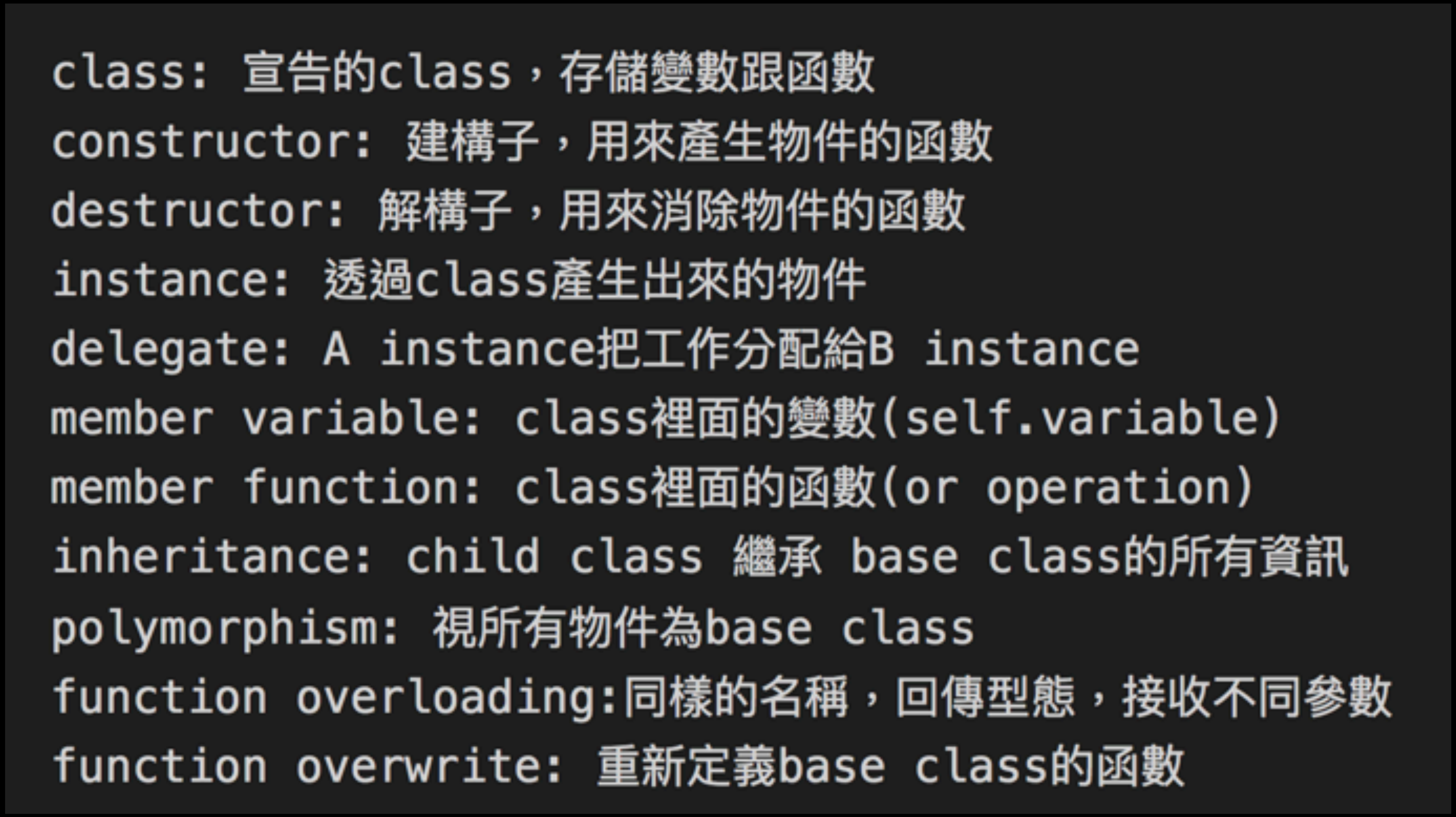
給予一個List裡面有溫度，把裡面所有的元素，轉成F，並寫入檔案

D2-Outline

1. What’s the OOP ?
2. OOP Term
3. Python Syntax
4. Inheritance
5. Polymorphism

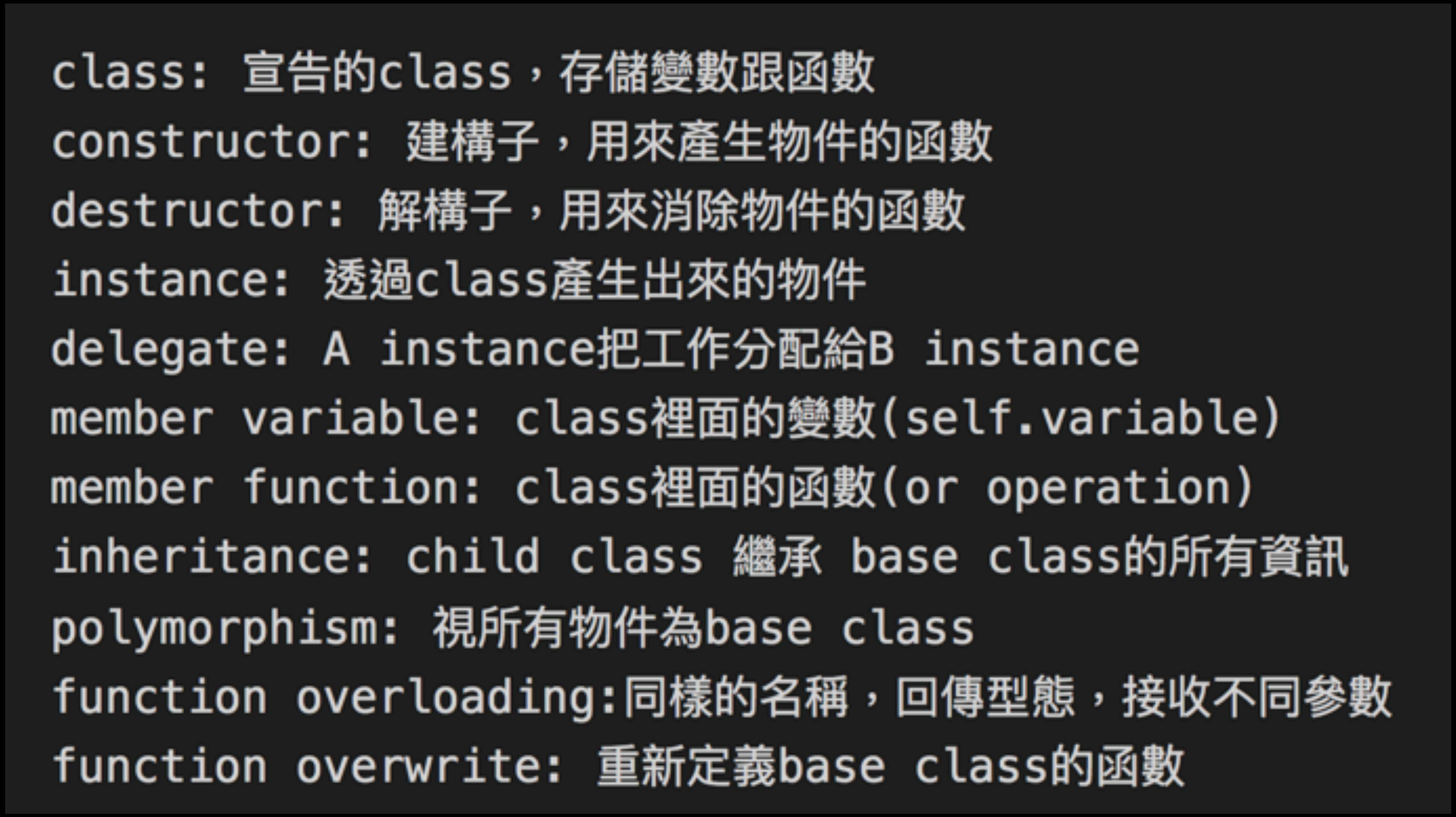
What’s The OOP ?

OOP Term



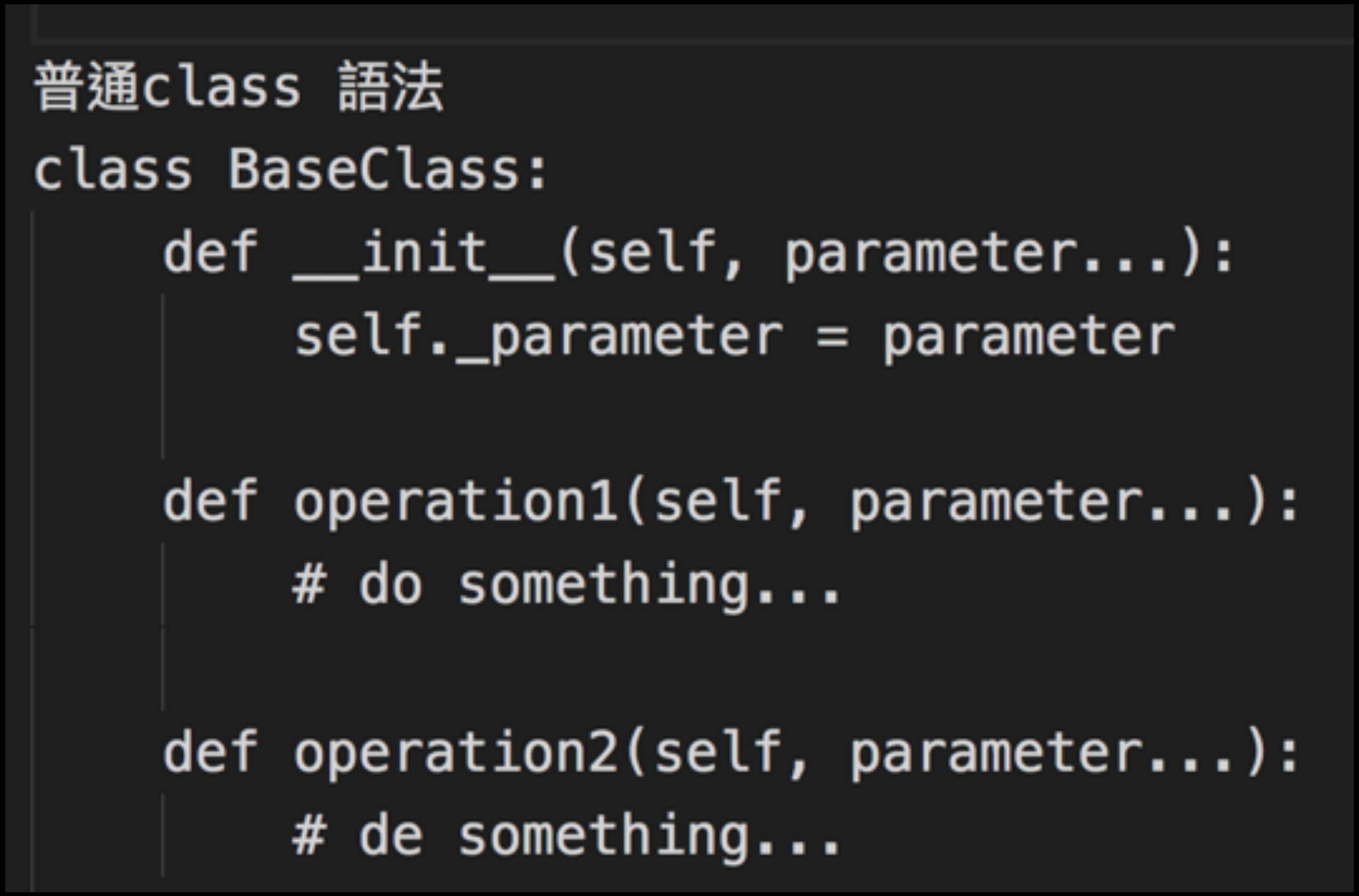
```
class: 宣告的class，存儲變數跟函數
constructor: 建構子，用來產生物件的函數
destructor: 解構子，用來消除物件的函數
instance: 透過class產生出來的物件
delegate: A instance把工作分配給B instance
member variable: class裡面的變數(self.variable)
member function: class裡面的函數(or operation)
inheritance: child class 繼承 base class的所有資訊
polymorphism: 視所有物件為base class
function overloading:同樣的名稱，回傳型態，接收不同參數
function overwrite: 重新定義base class的函數
```

OOP Term



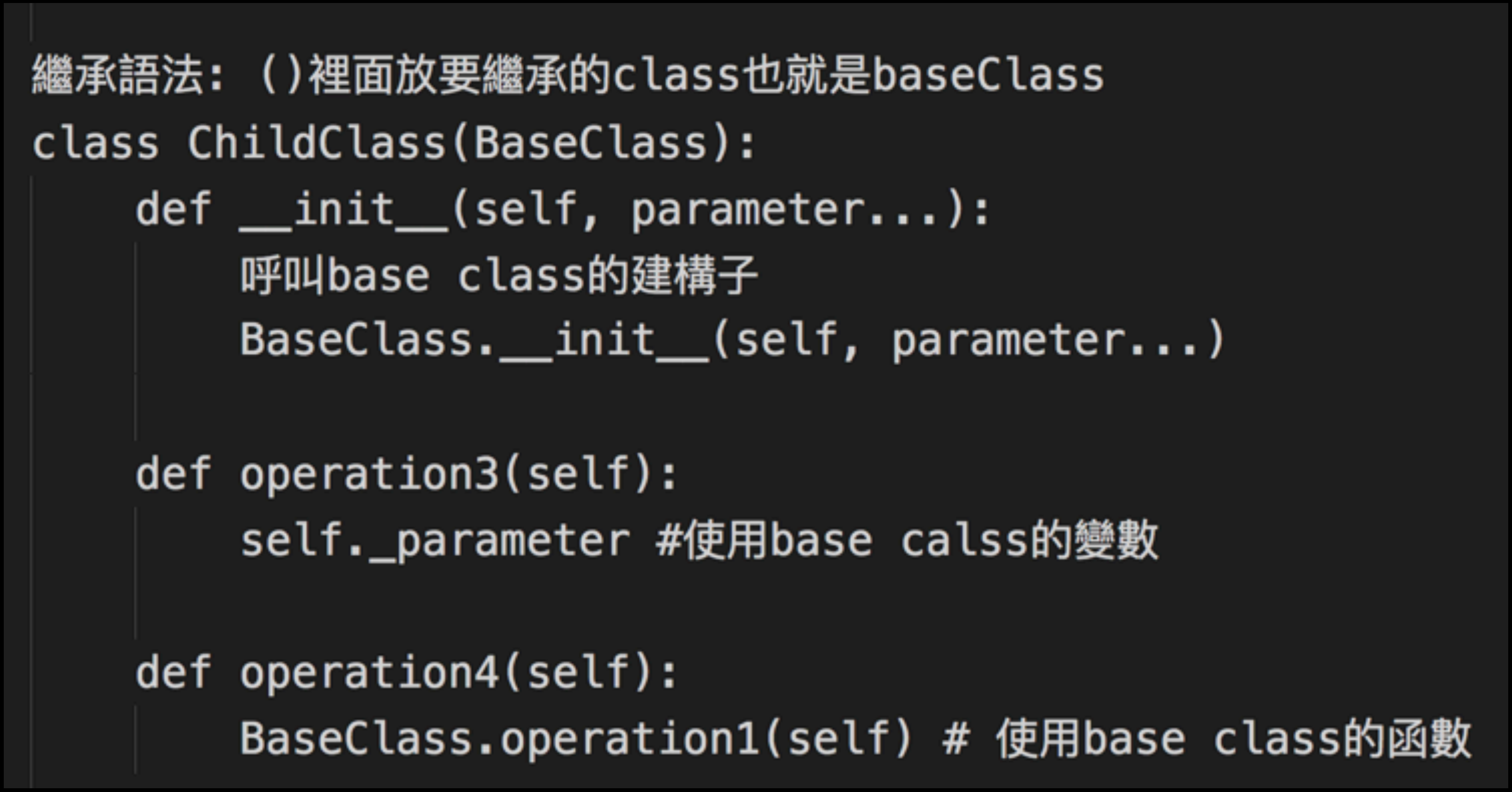
class: 宣告的class，存儲變數跟函數
constructor: 建構子，用來產生物件的函數
destructor: 解構子，用來消除物件的函數
instance: 透過class產生出來的物件
delegate: A instance把工作分配給B instance
member variable: class裡面的變數(self.variable)
member function: class裡面的函數(or operation)
inheritance: child class 繼承 base class的所有資訊
polymorphism: 視所有物件為base class
function overloading:同樣的名稱，回傳型態，接收不同參數
function overwrite: 重新定義base class的函數

Python Syntax



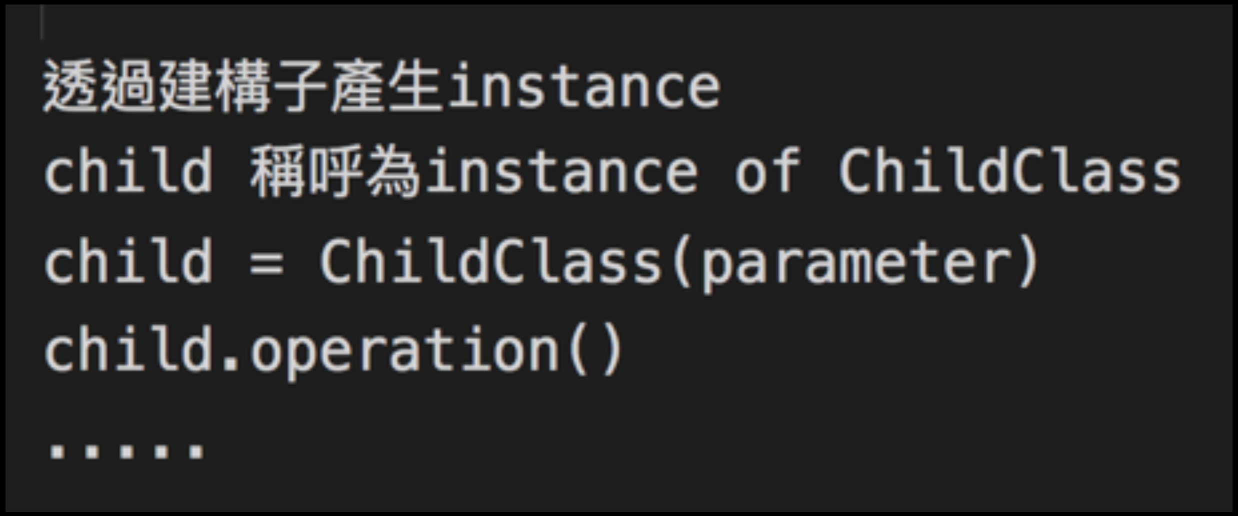
```
普通class 語法
class BaseClass:
    def __init__(self, parameter...):
        self._parameter = parameter

    def operation1(self, parameter...):
        # do something...

    def operation2(self, parameter...):
        # de something...
```

Python Syntax



```
繼承語法: ()裡面放要繼承的class也就是baseClass
class ChildClass(BaseClass):
    def __init__(self, parameter...):
        呼叫base class的建構子
        BaseClass.__init__(self, parameter...)

    def operation3(self):
        self._parameter #使用base calss的變數

    def operation4(self):
        BaseClass.operation1(self) # 使用base class的函數
```

Python Syntax



```
透過建構子產生instance
child 稱呼為instance of ChildClass
child = ChildClass(parameter)
child.operation()
.....
```

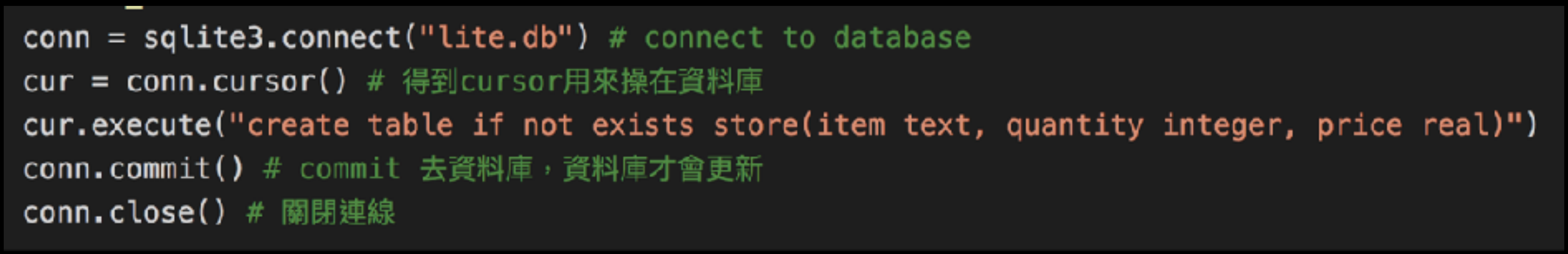
Inheritance DEMO

Polymorphism DEMO

D3-Outline

1. SQLite
2. Database Concept
3. SQL of SQLite

SQLite



```
conn = sqlite3.connect("lite.db") # connect to database
cur = conn.cursor() # 得到cursor用來操在資料庫
cur.execute("create table if not exists store(item text, quantity integer, price real)")
conn.commit() # commit 去資料庫，資料庫才會更新
conn.close() # 關閉連線
```

Database Concept

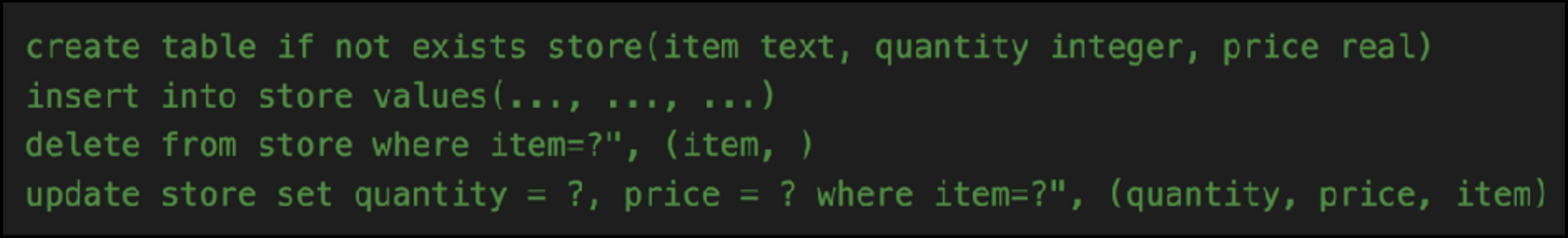
資料的形式用table的方式儲存，每個資料都有自己的型態

What’s The SQL?

Structured Query Language

對資料庫新增、修改、刪除，還有更進階的操作。

SQL

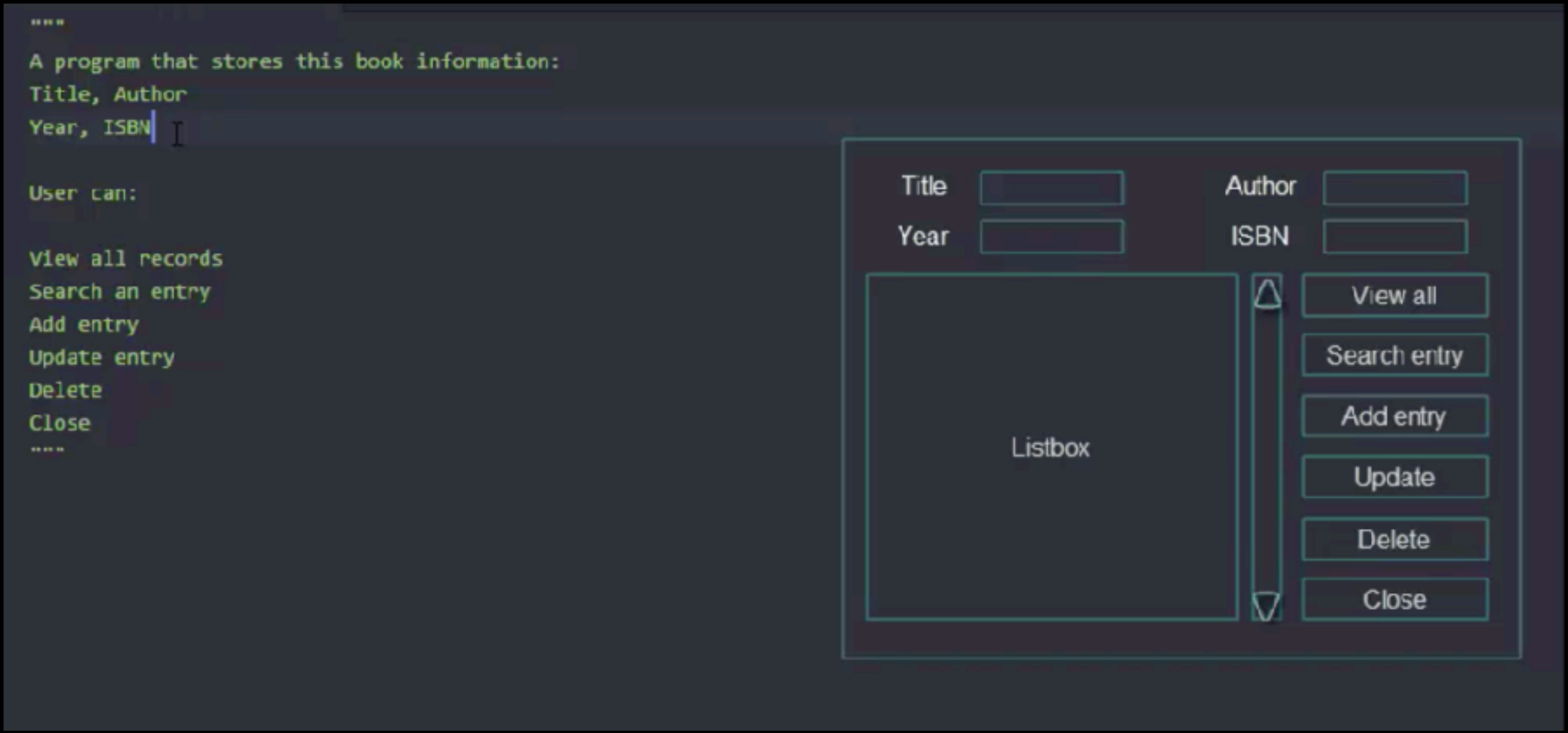


```
create table if not exists store(item text, quantity integer, price real)
insert into store values(..., ..., ...)
delete from store where item=?", (item, )
update store set quantity = ?, price = ? where item=?", (quantity, price, item)
```

D4-Outline

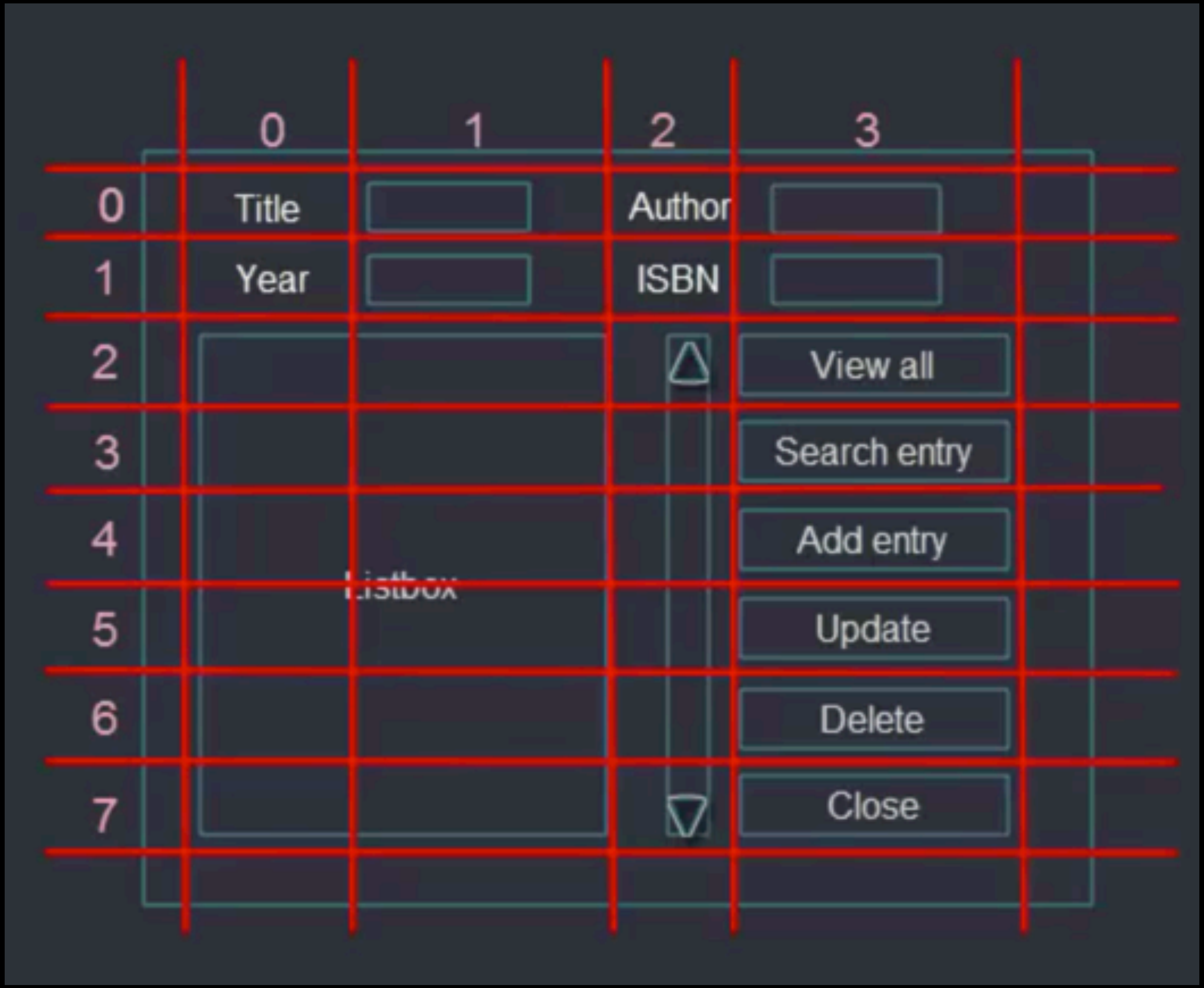
1. Tkinter
2. Desktop Project
3. Others

Tkinter

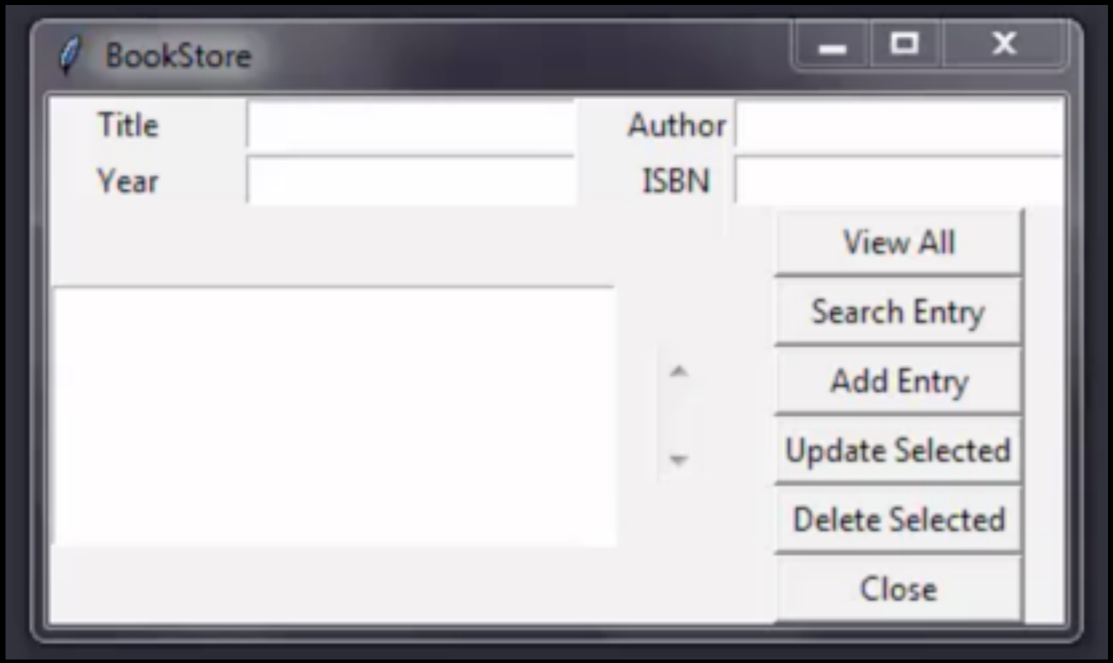


```
"""
A program that stores this book information:
Title, Author
Year, ISBN

User can:

View all records
Search an entry
Add entry
Update entry
Delete
Close
"""
```

Tkinter


0
1
2
3
0
1
2
3
4
5
6
7
Title
Author
Year
ISBN
Listbox
View all
Search entry
Add entry
Update
Delete
Close

Tkinter


BookStore
Title
Author
Year
ISBN
View All
Search Entry
Add Entry
Update Selected
Delete Selected
Close

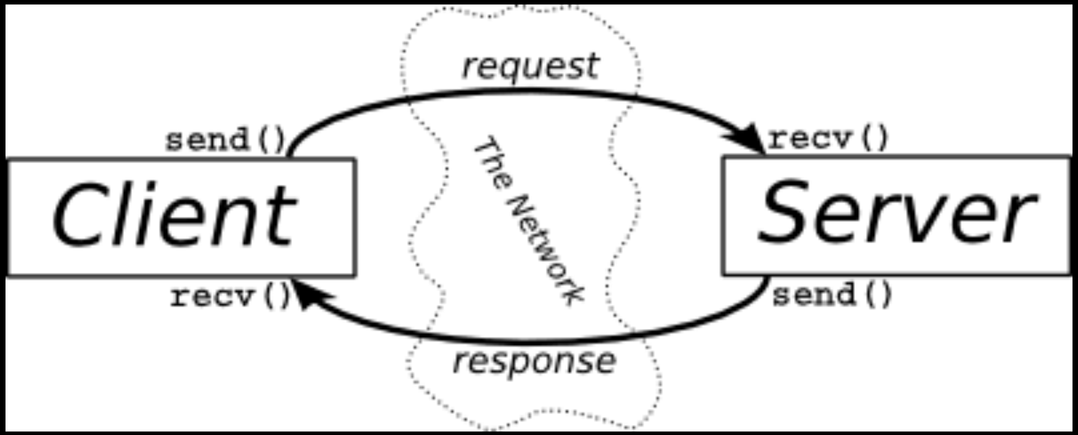
Desktop Project

Other

D5-Outline

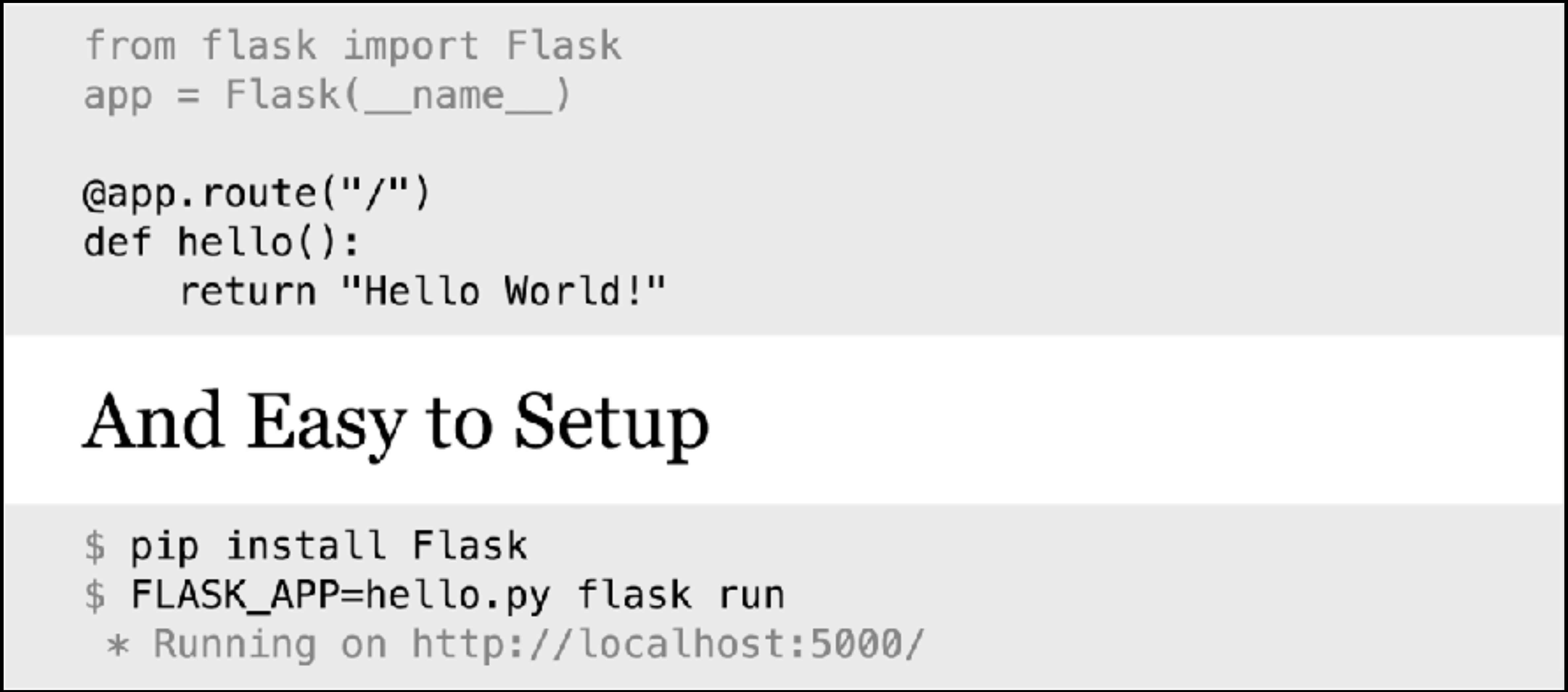
1. Web Concept
2. Python Flask
3. HTML
4. CSS
5. JAVASCRIPT

Web Concept



透過HTTP protocol

Flask

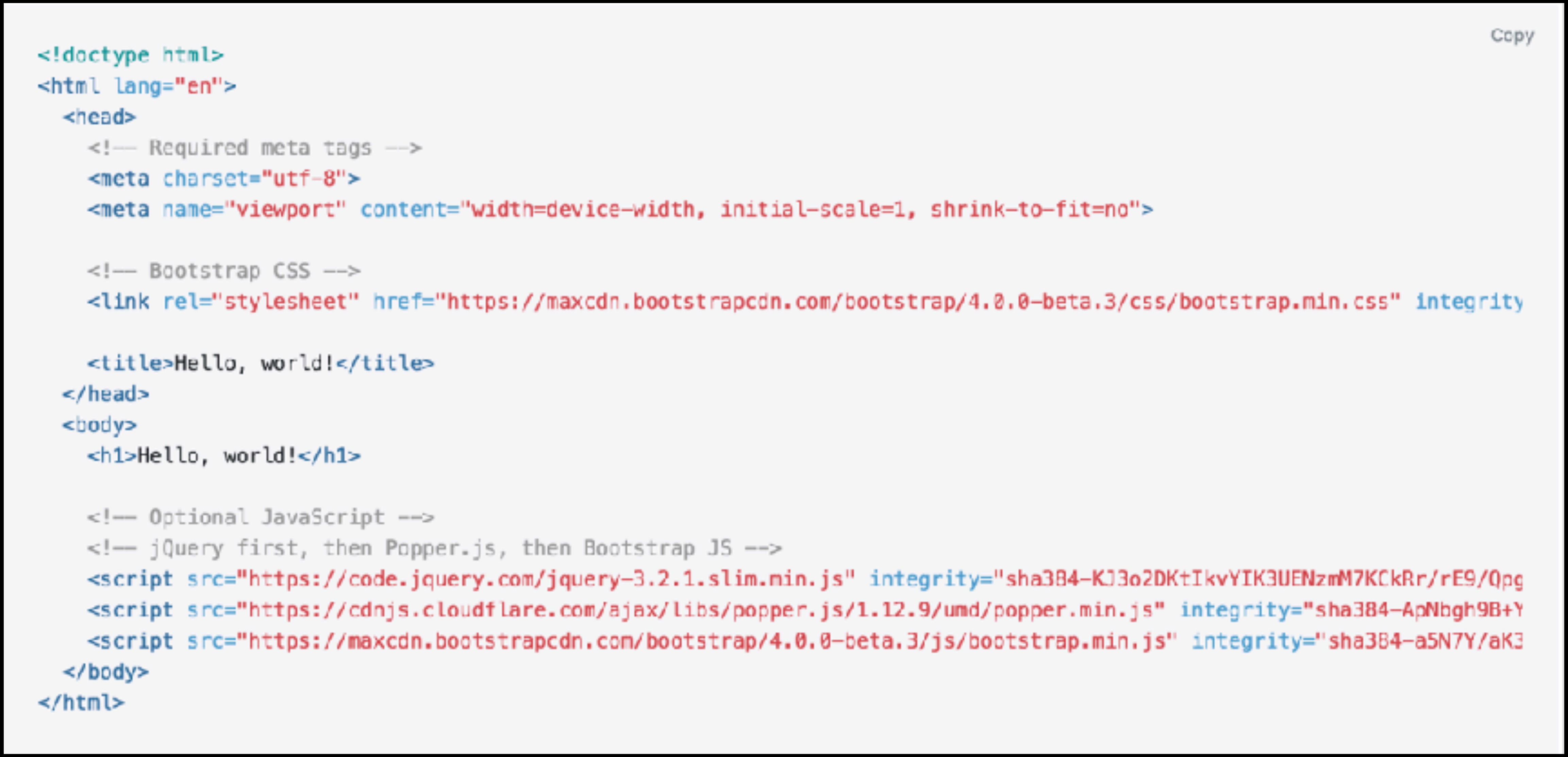


```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!"
```

And Easy to Setup

```
$ pip install Flask
$ FLASK_APP=hello.py flask run
 * Running on http://localhost:5000/
```

HTML


Copy

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.3/css/bootstrap.min.css" integrity

    <title>Hello, world!</title>
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.3/js/bootstrap.min.js" integrity="sha384-a5N7Y/aK3
  </body>
</html>
```

CSS

裝飾HTML，裝飾網頁外觀

JAVASCRIPT

處理網頁動態效果，跟使用者互動