PC教室電腦帳號/密碼:
clouder / cloud123

請打開桌面瀏覽器

首頁為本次上課之內容網頁

Python語言基礎

Why Python?

- 動態型別
- 內建物件型態
  - List, dictionary, string
- 內建的工具
  - Concatenation, slicing, sorting, mapping
- 記憶體自動管理
- 大型程式開發支援
  - Module, class (OOP), exception

物件型態與運算子(1/3)

- 數值

```
>>> width = 20
>>> height = 5*9
>>> width * height
900
```

- 字串

```
>>> 'spam eggs'
'spam eggs'
>>> 'doesn\'t'
"doesn't"
>>> "doesn't"
"doesn't"
>>>
"\"Yes,\" he said."
'"Yes," he said.‘
```

物件型態與運算子(2/3)

- List

```
>>> a = ['spam', 'eggs', 100, 1234]
>>> a
['spam', 'eggs', 100, 1234]

>>> a[0]
'spam'
>>> a[3]
1234
>>> a[-2] 100
```

物件型態與運算子(3/3)

- Dictionary

```
>>> tel = {'jack': 4098, 'sape': 4139}
>>> tel['guido'] = 4127
>>> tel
{'sape': 4139, 'guido': 4127, 'jack': 4098}

>>> tel['jack']
4098

>>> del tel['sape']
>>> tel['irv'] = 4127
>>> tel
{'guido': 4127, 'irv': 4127, 'jack': 4098}

>>> tel.keys()
['guido', 'irv', 'jack']
>>> 'guido' in tel
True
```

流程控制(1/3)

- if 表示式

```
>>> x = int(raw_input("Please enter an integer: "))
Please enter an integer: 42
>>> if x < 0:
...         x = 0
...         print 'Negative changed to zero'
...     elif x == 0:
...         print 'Zero'
...     elif x == 1:
...         print 'Single'
...     else:
...         print 'More'
...
More
```

流程控制(2/3)

- for 表示式

```
>>> # Measure some strings:
...     a = ['cat', 'window', 'defenestrate']
>>> for x in a:
...         print x, len(x)
...
cat 3
window 6
defenestrate
```

- while表示式

```
>>> while True:
...         pass        # Busy-wait for keyboard interrupt (Ctrl+C)
>>> # Fibonacci series:
...     a, b = 0, 1
>>> while b < 10:
...         print b
...         a, b = b, a+b
```

流程控制(3/3)

- range 表示式

```
>>> range(10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

>>> range(5, 10)
[5, 6, 7, 8, 9]

>>> range(0, 10, 3)
[0, 3, 6, 9]

>>> a = ['Mary', 'had', 'a', 'little', 'lamb']
>>> for i in range(len(a)):
...         print i, a[i]
0 Mary
1 had
2 a
3 little
4 lamb
```

函式與模組(1/2)

- 定義function

```
>>> def fib(n):    # write Fibonacci series up to n
...     """Print a Fibonacci series up to n."""
...     a, b = 0, 1
...     while a < n:
...         print a,
...         a, b = b, a+b
...
>>> # Now call the function we just defined:
... fib(2000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597
```

函式與模組(2/2)

- 自訂模組

fibo.py

```
# Fibonacci numbers module
def fib_function(n):    # write Fibonacci series up to n
    a, b = 0, 1
    while b < n:
        print b, a, b = b, a+b
```

```
>>> import fibo
>>> fibo.fib_function(1000)
1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
>>> fibo.__name__
'fibo'
```

```
from fibo import *  (或fibo_function)
```

類別(1/2)

- 類別宣告

```
class MyClass:
    """A simple example class"""
    i = 12345
    def f(self):
        return ‘hello world‘
```

- 初始化

```
>>> class Complex:
...         def __init__(self, realpart, imagpart):
...             self.r = realpart
...             self.i = imagpart
...
>>> x = Complex(3.0, -4.5)
>>> x.r, x.i
(3.0, -4.5)
```

類別(2/2)

- 繼承

  多重繼承：

```
class MyClass(FatherClass, MotherClass):
    """A inheritance example class"""
    i = 12345
    def f(self):
        return ' Inheritance Hello!'
```