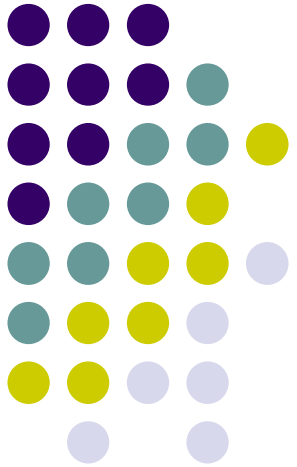
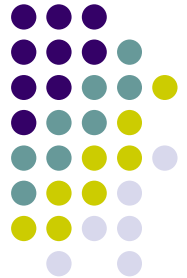
Python程式設計入門
容器-序列(2/2)

葉難

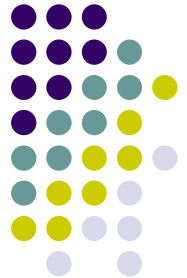


大綱



- Sequence（序列）抽象型別
- list（串列）， tuple（元組）， str（字串）
- 迭代（iteration）協定、可迭代者（Iterable）、迭代器（Iterator）
- 串列生成式（list comprehension）
- 產生器運算式（generator expression）

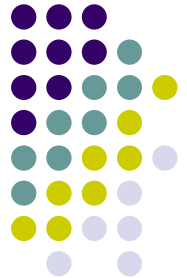
串列生成式（**list comprehension**）



- 屬於「運算式」，簡稱listcomp
- Python語言以list爲主力資料結構，相關函式、方法、語法非常多
- 簡單的迴圈程式，都能改爲生成式
- 語法

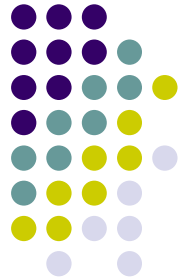
```
[ 運算式 for 名稱 in 可迭代者 ]
[ 運算式 for 名稱 in 可迭代者 if 運算式]
```

華氏溫度轉攝氏：迴圈與生成式



```
def ftoc(ft):
    ct = []
    for x in ft:
        ct.append((x - 32) * 5 / 9)
    return ct
####
ft = [32, 212, 10, 55, 78]
ct = [(x-32)*5/9 for x in ft]
```

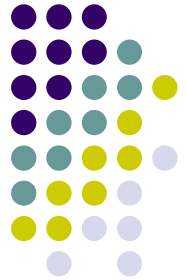
串列生成式，不是述句



- 是「運算式」，結果是個list物件
- 任何可放運算式的地方，都可放listcomp

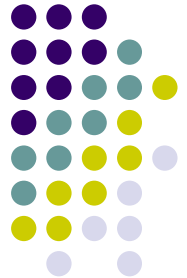
```
for x in [(x-32)*5/9 for x in ft]:
    print(x)

li = [[x*2 for x in range(5)]]
li.append([x for x in range(5)])
```

範例



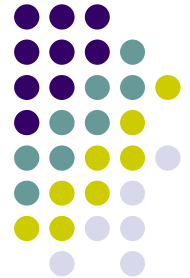
```
>>> [(t, 331+t*0.6) for t in range(20, 40, 5)]
[(20, 343.0), (25, 346.0), (30, 349.0), (35,
  352.0)]          # 溫度 與 音速

>>> from math import sqrt
>>> scores = [64, 35, 59, 100, 91, 87] # 成績
>>> [int(sqrt(x) * 10) for x in scores]
[80, 59, 74, 76, 100, 95, 93] # 調整後的成績
```

範例：加上if



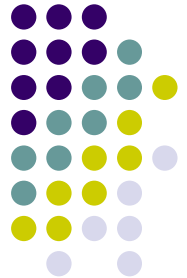
```
>>> li = [20, 15, 3100, 599, 75]  # 消費金額
>>> li2 = [x for x in li if x <= 1000]
>>> sum(li2) / len(li2)   # 小額消費的平均金額
177.25
>>> fs = ['a.jpg', 'b.bmp', 'c.py', 'd.jpg']
>>> [x for x in fs if x[-4:] == '.jpg']
['a.jpg', 'd.jpg']
>>> scores = [64, 35, 59, 100, 91, 87]  # 成績
>>> sum([1 for x in scores if x >= 60])
4    # PASS人數
```

問題：3或5的倍數



- 小於10、且是3或5的倍數的數字，有3、5、6、9，總和是23。
- 請問小於1000、 且是3或5的倍數的數字，總和是多少？
- 請分別用迴圈和串列生成式撰寫

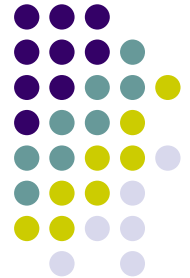
問題：含位元資料的字串轉int



- 字串內容含位元資料（0與1），長度是8的倍數
- 每8個視爲一個byte，轉成int物件
- 例：

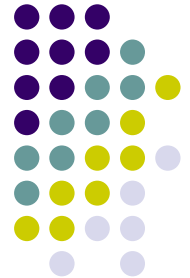
```
data1 = '0011010100001111' # 16個位元
# 應轉成[53, 15]

data2 = '0001101001000000011001001100100011111110'
# 轉成結果是？
```

串列生成式：雙重，九九乘法



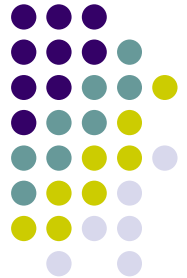
```
result = []
for x in range(2, 9+1)
    for y in range(1, 9+1)
        result.append(str(x) + '*' + str(y) +
                                '=' + str(x*y))

####
result = [str(x)+'*'+str(y)+'='+str(x*y)
                for x in range(2, 9+1)
                    for y in range(1, 9+1)]
```

雙重再加上if：組合和過濾



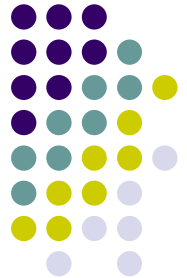
```
names = ['a', 'b', 'c']
exts = ['.jpg', '.png', '.bmp']
result = []
for x in names:
    if x != 'a':
        for y in exts:
            if y != '.jpg':
                result.append(x + y)
####
result = [x+y for x in names if x!='a'
                  for y in exts if y!= '.jpg']
```

3.x版的print是函式，屬於運算式



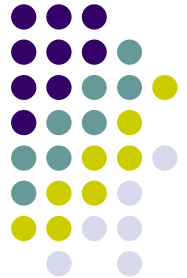
```
>>> from math import sqrt
>>> [print(sqrt(x)) for x in range(5)]
0.0
1.0
1.4142135623730951
1.7320508075688772
2.0
[None, None, None, None, None]
```

不要濫用



```
images = ['green_D.jpg', 'red_D.gif', 'orange_R.jpg',
  'black_S.gif', 'folder_A', 'folder_B']
exts = ['jpg', 'bmp', 'png', 'gif']
excluded = ['_R.', '_A.', '_S.']
```

- 挑出副檔名符合exts，但檔名不包含excluded
- `[img for img in images if any([img.endswith(ext) for ext in exts]) and not any([bad in img for bad in excluded])]`

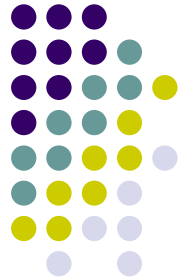


問題：

- 分數與加權，請算出加權總分

```
scores = (60, 70)
weights = (3, 4)
# 加權總分是60*3 + 70*4，得到460

scores = (30, 60, 50, 80, 95)
weights = (3, 4, 2, 4, 3)
# 加權總分是多少？
```

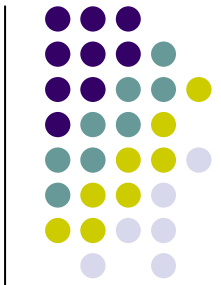
問題：複利



- 假設一開始存款10萬，複利2%，請列出第一年到第十年的存款數目

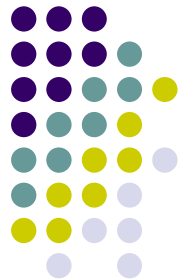
```
>>> x = 10; y = 1.02
>>> (x * y**1) * 10000
102000.0
>>> (x * y**2) * 10000
104040.0
# 請寫listcomp，算出如下結果
[102000, 104040, 106120, 108243, 110408,
  112616, 114868, 117165, 119509, 121899]
```

產生器（generator）



- 產生器運算式
- 產生器函式，yield述句
- 產生器物件符合迭代器介面
- 產生器：
    1. 實作Iterator的簡便方式
    2. 支援委託（delegation）概念：yield from
    3. coroutine：send

產生器運算式（generator expression）



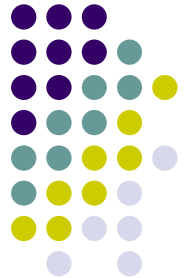
- 語法，與listcomp相同，但使用小括號

```
( 運算式 for 名稱 in 可迭代者 )
( 運算式 for 名稱 in 可迭代者 if 運算式 )
```

- 其結果是產生器物件，符合迭代器介面
- 若作爲單一參數傳入函式，可省略小括號

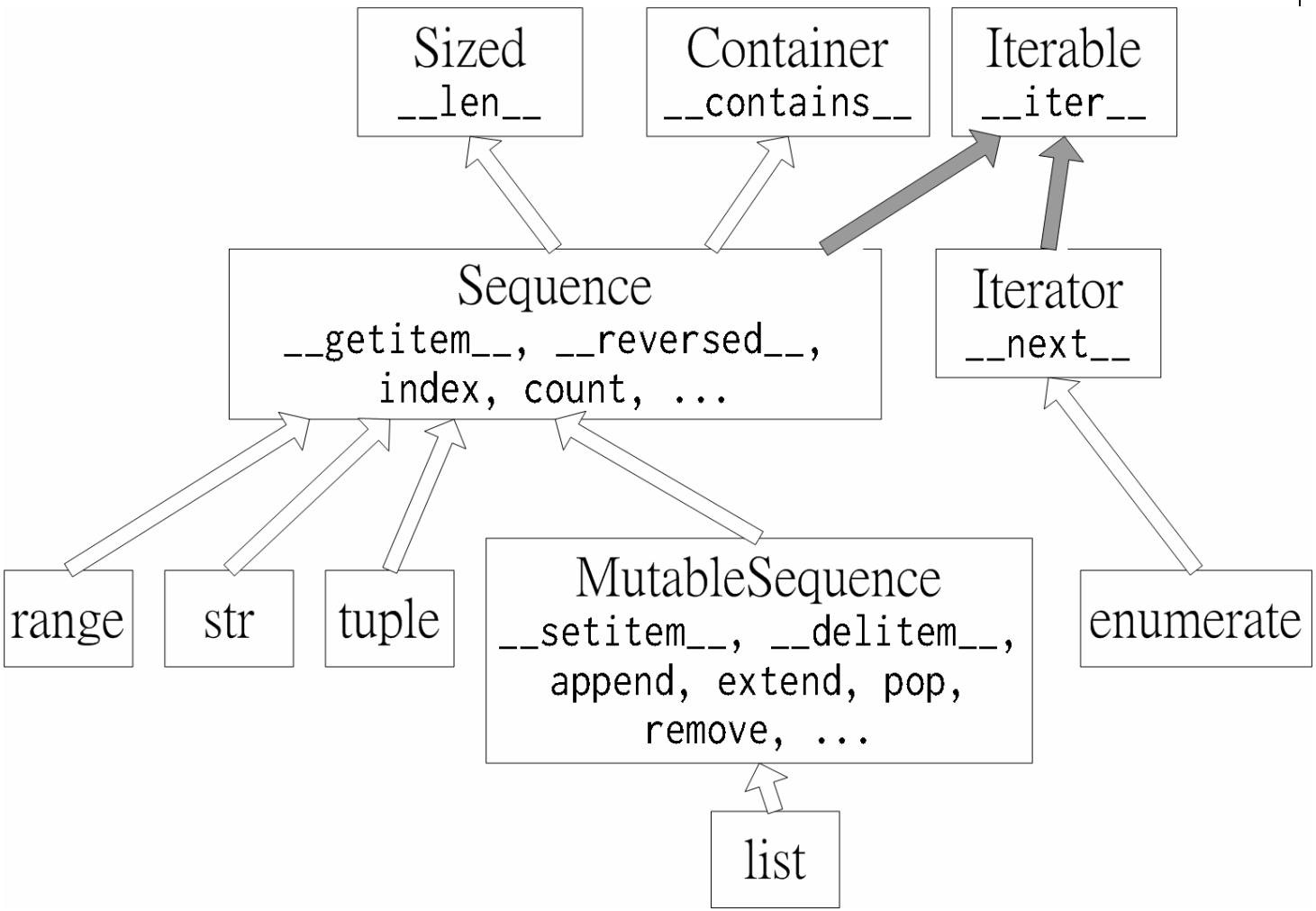
```
>>> (x**2 for x in range(10))
<generator object <genexpr> at 0x0049B850>
```

產生器物件是迭代器

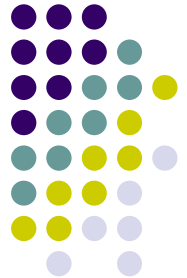


```
>>> g = (x**2 for x in range(3))
>>> iter(g) is g
True
>>> next(g)
0
>>> next(g), next(g)
(1, 4)
>>> next(g)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
StopIteration
```

迭代相關抽象型別


Sized
__len__
Container
__contains__
Iterable
__iter__
Sequence
__getitem__, __reversed__,
index, count, ...
Iterator
__next__
range
str
tuple
MutableSequence
__setitem__, __delitem__,
append, extend, pop,
remove, ...
enumerate
list

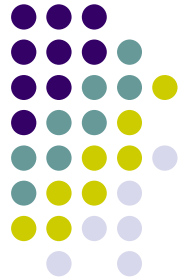
可省略小括號的情況



- 若作爲單一參數傳入函式時

```
>>> sum( [ x**2 for x in range(10) ] )
285
>>> sum( ( x**2 for x in range(10) ) )
285
>>> sum(  x**2 for x in range(10)  )
285
>>> vx = (1, 2, 3); vy = (2, 4, 6)
>>> sum( x*y for x, y in zip(vx, vy) )
28
```

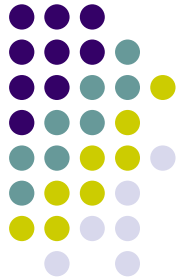
函式參數要求的型別



- Sequence或Iterable

```
>>> sorted(x**2 for x in range(4))
[0, 1, 4, 9]
>>> enumerate(x**2 for x in range(4))
<enumerate object at 0x0049B6E8>
>>> reversed(x**2 for x in range(4))
TypeError: argument to reversed() must be a sequence
```

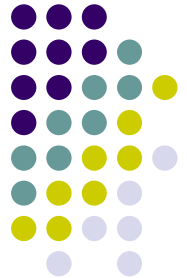
問題



- 男生標準體重：(身高 - 80) * 0.7
- 女生標準體重：(身高 - 70) * 0.6
- 請算出（體重 減 標準體重）的平均值

```
data = (('Amy', 'female', 160, 65),
        ('Bob', 'male', 180, 83),
        ('Cathy', 'female', 172, 66),
        ('David', 'male', 177, 92))
print(sum( ??? ) / len(data))
```

以產生器運算式重寫MyDeck



- 亂數給出52張牌

```
import random
suits = ('Spade', 'Heart', 'Diamond', 'Club')
ranks = ('A', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'J', 'Q', 'K')

deck = (suits[n//13] + ' ' + ranks[n%13]
    for n in random.sample(range(52), 52))

for _ in range(5):
    print(next(deck))
```

Q&A