Python
List Comprehensions

Stéphane Vialette

LIGM, Université Paris-Est Marne-la-Vallée

October 9, 2009

List comprehensions

Description

- List comprehensions (or 'list comps' for short) come from the functional programming language Haskell.
- Simple and flexible utility tool that help to create lists on the fly.
- Added to Python in version 2.0.
- A clear a valuable alternative to `map` and `filter`
- Similar to set definitions

$$S = \{x : x \in \mathbb{N} \text{ and } x^2 \leq 120\}$$

Introducing list comprehensions

Example

```
In [1]: s = range(10)

In [2]: print s
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

In [3]: print [x*x for x in s]
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]

In [4]: print [x*x for x in s if x <= 4]
[0, 1, 4, 9, 16]

In [5]: print [x*x for x in s if x % 2 == 0]
[0, 4, 16, 36, 64]

In [6]: print [(x, x*x) for x in s if x % 2 == 0]
[(0, 0), (2, 4), (4, 16), (6, 36), (8, 64)]
```

Introducing list comprehensions

Example

```
In [1]: s = range(10)

In [2]: print [x for x in s if x > 3 and x < 7]
[4, 5, 6]

In [3]: print [x for x in s if x > 3 and x < 7 and x % 2 == 0]
[4, 6]

In [4]: fruits = ['ananas', 'apple', 'orange', 'pineapple']

In [5]: print [fruit for fruit in fruits if 'e' in fruit]
['apple', 'orange', 'pineapple']

In [6]: print [fruit for fruit in fruits if len(fruit) <= 6]
['ananas', 'apple', 'orange']

In [7]: print [fruit for fruit in fruits if 'e' in fruit and len(fruit) <= 6]
['apple', 'orange']
```

Introducing list comprehensions

Example

```
In [1]: s1, s2 = range(4), ['a', 'b', 'c', 'd']

In [2]: print [(x, y) for x in s1 for y in s2]
[(0, 'a'), (0, 'b'), (0, 'c'), (0, 'd'), (1, 'a'), (1, 'b'), (1, 'c'), (1, 'd'), (2, 'a'), (2, 'b'),
 (2, 'c'), (2, 'd'), (3, 'a'), (3, 'b'), (3, 'c'), (3, 'd')]

In [3]: print [(x, y) for x in s1 for y in s2 if x < 3]
[(0, 'a'), (0, 'b'), (0, 'c'), (0, 'd'), (1, 'a'), (1, 'b'), (1, 'c'), (1, 'd'), (2, 'a'), (2, 'b'),
 (2, 'c'), (2, 'd')]

In [4]: print [(x, y) for x in s1 for y in s2 if x < 3 and (y == 'a' or y == 'c')]
[(0, 'a'), (0, 'c'), (1, 'a'), (1, 'c'), (2, 'a'), (2, 'c')]

In [5]: s1, s2 = range(4), range(10,14)

In [6]: print [(x, y) for x in s1 for y in s2 if x % 2 == y % 2 and x < 3]
[(0, 10), (0, 12), (1, 11), (1, 13), (2, 10), (2, 12)]
```

Introducing list comprehensions

Example

```
In [1]: s = range(1000)

In [2]: print [x for x in s if x*x in s]
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27,
28, 29, 30, 31]

In [3]: import math

In [4]: print [x for x in s if x*x in s and math.sqrt(x) in s]
[0, 1, 4, 9, 16, 25]

In [5]: print [(x, y) for x in s for y in s if x*x == y and 4 <= y <= 20]
[(2, 4), (3, 9), (4, 16)]

In [6]: print [(x, y) for x in s for y in s if x*x == y and 4 <= y <= 20 and x % 2]
[(3, 9)]
```

Introducing list comprehensions

Example

```
In [1]: s = range(1, 100)

In [2]: np = [x for x in s for y in s for z in s if x == y*z and y != x and z != x]

In [3]: p = [x for x in s if x not in np]

In [4]: print p
[1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]
```

Quicksort

Example

```
In [1]: def qsort(s):
   ...:     if len(s) == 0:
   ...:         return []
   ...:     else:
   ...:         return qsort([x for x in s[1:] if x < s[0]]) + \
                       [s[0]]                                + \
                       qsort([x for x in s[1:] if x >= s[0]])
   ...:
In [2]: print qsort([])
[]

In [3]: print qsort([2, 4, 1, 3, 5, 8, 6, 7])
[1, 2, 3, 4, 5, 6, 7, 8]

In [4]: print qsort('ananas')
['a', 'a', 'a', 'n', 'n', 's']

In [5]: print ''.join(qsort('ananas'))
aaanns
```