

2014-04-09.sagews

April 9, 2014

Contents

1	Math 480b Sage Course	1
1.1	April 9, 2014	1
1.2	Python: List Comprehensions	1
1.3	Python Functions	5
1.4	Putting your code in a file	6

TOTAL : 13/20

1 Math 480b Sage Course

1.1 April 9, 2014

Screencast: <http://youtu.be/ndoM3TJGSDY>
Plan

- Questions?
- Python
 - list comprehensions
 - functions
 - putting code in files/modules and loading them
- Friday: creating your own data types

1.2 Python: List Comprehensions

- simple examples turning a for loop to make a list into a comprehension
- more complicated examples
- map versus list comprehension
- the scope gotcha

```
v = []
for n in range(5):
    v.append((n*(n+1))/2)
```

```
v
[0, 1, 3, 6, 10]
```

```
v = [n*(n+1)/2 for n in range(5)]
v
[0, 1, 3, 6, 10]
```

```
v = []
for n in range(5):
    c = n*(n+1)/2
    if c % 2 == 0:
        v.append(c)
v
[0, 6, 10]
```

```
v = [n*(n+1)/2 for n in range(5) if n*(n+1)/2 %2 == 0]
v
[0, 6, 10]
```

```
v = []
for n in range(1,5):
    for m in range(5,9):
        v.append(n*m)
v
[5, 6, 7, 8, 10, 12, 14, 16, 15, 18, 21, 24, 20, 24, 28, 32]
```

```
[n*m for n in range(1,5) for m in range(5,9)]
[5, 6, 7, 8, 10, 12, 14, 16, 15, 18, 21, 24, 20, 24, 28, 32]
```

```
x = 784
```

```
[N(sin(x)) for x in range(1,10)]
[0.841470984807897, 0.909297426825682, 0.141120008059867, -0.756802495307928,
-0.958924274663138, -0.279415498198926, 0.656986598718789, 0.989358246623382,
0.412118485241757]
```

```
x
9
```

```
[sin(x*pi/3) for x in range(1,10)]
[1/2*sqrt(3), 1/2*sqrt(3), 0, -1/2*sqrt(3), -1/2*sqrt(3), 0, 1/2*sqrt(3), 1/2*sqrt(3), 0]
```

```
[math.sin(x*pi/3) for x in range(1,10)]
[0.8660254037844386, 0.8660254037844387, 1.2246467991473532e-16, -0.8660254037844384,
-0.8660254037844386, -2.4492935982947064e-16, 0.8660254037844388, 0.8660254037844392,
3.6739403974420594e-16]
```

```
[n*2 for n in [k^2 for k in range(4)]]
[0, 2, 8, 18]
```

```
%time
v = []
for n in range(10^6):
    if n % 2 == 1:
        v.append(n*n)
CPU time: 4.60 s, Wall time: 4.60 s
```

```
%time
v = [n*n for n in range(10^6) if n %2 == 1]
CPU time: 3.06 s, Wall time: 3.05 s
```

```
[1..5]
[1, 2, 3, 4, 5]
```

```
v = [n*n for n in [1..10^6] if n %2 == 1]
```

```
[1,5,..,100]
[1, 5, 9, 13, 17, 21, 25, 29, 33, 37, 41, 45, 49, 53, 57, 61, 65, 69, 73, 77, 81, 85, 89,
93, 97]
```

```
show([1,5+pi,..100])
[1,  $\pi + 5$ ,  $2\pi + 9$ ,  $3\pi + 13$ ,  $4\pi + 17$ ,  $5\pi + 21$ ,  $6\pi + 25$ ,  $7\pi + 29$ ,  $8\pi + 33$ ,  $9\pi + 37$ ,  $10\pi + 41$ ,  $11\pi + 45$ ,  $12\pi + 49$ ,  $13\pi + 53$ ]
```

```
%var x
s = expand((1+sqrt(2)*x+e^pi)^3); s
2*sqrt(2)*x^3 + 6*x^2*e^pi + 6*sqrt(2)*x*e^pi + 3*sqrt(2)*x*e^(2*pi) + 6*x^2 + 3*sqrt(2)*x
+ 3*e^pi + e^(3*pi) + 3*e^(2*pi) + 1
```

```
show(s)

$$2\sqrt{2}x^3 + 6x^2e^\pi + 6\sqrt{2}xe^\pi + 3\sqrt{2}xe^{(2\pi)} + 6x^2 + 3\sqrt{2}x + 3e^\pi + e^{(3\pi)} + 3e^{(2\pi)} + 1$$

```

```
latex(s)
2 \, , \sqrt{2} x^{\{3\}} + 6 \, , x^{\{2\}} e^{\{\pi\}} + 6 \, , \sqrt{2} x e^{\{\pi\}} + 3 \, , \sqrt{2} x
e^{\{\left(2 \, , \pi\right)\}} + 6 \, , x^{\{2\}} + 3 \, , \sqrt{2} x + 3 \, , e^{\{\pi\}} + e^{\{\left(3 \, ,
\pi\right)\}} + 3 \, , e^{\{\left(2 \, , \pi\right)\}} + 1
```

```
v = (k^3 for k in xrange(10^10))
```

```
v
<generator object <genexpr> at 0x7472190>
```

```
for n in v:
    print n+2
    if n >10:
        break
```

```
2
3
10
29
```

map?

Unable to read source filename (<built-in function map> is not a module, class, method, function, traceback, frame, or code object) Docstring:
map(function, sequence[, sequence, ...]) -> list

Return a list of the results of applying the function to the items of the argument sequence(s). If more than one sequence is given, the function is called with an argument list consisting of the corresponding item of each sequence, substituting None for missing values when not all sequences have the same length. If the function is None, return a list of the items of the sequence (or a list of tuples if more than one sequence).

```
v = range(10)
[n*(n+1)/2 for n in v]
[0, 1, 3, 6, 10, 15, 21, 28, 36, 45]
```

```
def f(n):
    return n*(n+1)/2
map(f, v)
[0, 1, 3, 6, 10, 15, 21, 28, 36, 45]
```

```
map(lambda n : n*(n+1)/2, v)
[0, 1, 3, 6, 10, 15, 21, 28, 36, 45]
```

filter?

Unable to read source filename (<built-in function filter> is not a module, class, method, function, traceback, frame, or code object) Docstring:
filter(function or None, sequence) -> list, tuple, or string

Return those items of sequence for which function(item) is true. If function is None, return the items that are true. If sequence is a tuple or string, return the same type, else return a list.

reduce?

Unable to read source filename (<built-in function reduce> is not a module, class, method, function, traceback, frame, or code object) Docstring:
reduce(function, sequence[, initial]) -> value

Apply a function of two arguments cumulatively to the items of a sequence, from left to right, so as to reduce the sequence to a single value. For example, reduce(lambda x, y: x+y, [1, 2, 3, 4, 5]) calculates (((1+2)+3)+4)+5). If initial is present, it is placed before the items of the sequence in the calculation, and serves as a default when the sequence is empty.

```
# in magma...
&+[1..5]
```

1.3 Python Functions

- arguments are references (and the default argument gotcha)
- all functions return exactly one thing
- tuple packing/unpacking
- variable number of arguments and keywords
- local/global scope

```
def f(v, w=[]):
    print id(w)
    w.append(v)
    print v, w
```

```
f(2)
218375448
2 [2]
```

```
f(2)
218375448
2 [2, 2]
```

```
f(pi)
218375448
pi [2, 2, pi]
```

```
f(2)
2 []
```

```
# fix
```

```
def f(v, w=None):
    if w is None:
        w = []
    print id(w)
    w.append(v)
    print v, w
```

```
f(2)
218375736
2 [2]
```

```
f(2)
218376024
2 [2]
```

```
w = [1,2,3]
f(v=pi, w=w)
pi [1, 2, 3, pi]
```

```
w
[1, 2, 3, pi]
```

```
def length(v):
    v.append('he he')
    return len(v)-1
```

```
length(w)
4
```

```
w
[1, 2, 3, pi, 'he he']
```

1.4 Putting your code in a file

- loading .py, .sage files
- importing (and reloading) .py modules