NPTEL MOOC

PROGRAMMING, DATA STRUCTURES AND ALGORITHMS IN PYTHON

Week 4, Lecture 7

Madhavan Mukund, Chennai Mathematical Institute
http://www.cmi.ac.in/~madhavan

Operating on lists

- Update an entire list

```
for x in l:
    x = f(x)
```

- Define a function to do this in general

```
def applylist(f,l):
    for x in l:
        x = f(x)
```

NPTEL

Built in function `map()`

- `map(f,l)` applies `f` to each element of `l`
- Output of `map(f,l)` is not a list!
- Use `list(map(f,l))` to get a list
- Can be used directly in a `for` loop

```
for i in map(f,l):
```

- Like `range(i,j)`, `d.keys()`

Selecting a sublist

- Extract list of primes from list `numberlist`

```
primelist = []
for i in numberlist:
   if isprime(i):
      primelist.append(i)
return(primelist)
```

Selecting a sublist

- In general

```
def select(property,l):
    sublist = []
    for x in l:
        if property(x):
            sublist.append(x)
    return(sublist)
```

- Note that `property` is a function that returns `True` or `False` for each element

Built in function `filter()`

- `filter(p,l)` checks `p` for each element of `l`
- Output is sublist of values that satisfy `p`

Combining map and filter

- Squares of even numbers from 0 to 99

```
list(map(square,filter(iseven,range(100))

def square(x):
  return(x*x)

def iseven(x):
  return(x%2 == 0)
```

NPTEL

List comprehension

- Pythagorean triple: $x^2 + y^2 = z^2$
- All Pythagorean triples (x,y,z) with values below n

$$\{ (x,y,z) \mid 1 \leq x,y,z \leq n, x^2 + y^2 = z^2 \}$$

- In set theory, this is called set comprehension
  - Building a new set from existing sets
- Extend to lists

List comprehension

- Squares of even numbers below 100

```
[square(x) for i in range(100) if iseven(x)]
```

map generator filter

Multiple generators

- Pythagorean triples with x,y,z below 100

```
[(x,y,z) for x in range(100)
         for y in range(100)
         for z in range(100)
          if x*x + y*y == z*z]
```

- Order of x,y,z is like nested for loop

```
for x in range(100):
  for y in range(100):
    for z in range(100):
```

Multiple generators

- Later generators can depend on earlier ones
- Pythagorean triples with x,y,z below 100, no duplicates

```
[(x,y,z) for x in range(100)
         for y in range(x,100)
          for z in range(y,100)
           if x*x + y*y == z*z]
```

Useful for initialising lists

- Initialise a 4 x 3 matrix
  - 4 rows, 3 columns
- Stored row-wise

```
l = [ [ 0 for i in range(3) ]
            for j in range(4)]
```

NPTEL

Warning

- What's happening here?

```
>>> zerolist = [ 0 for i in range(3) ]
>>> l = [ zerolist for j in range(4) ]

>>> l[1][1] = 7

>>> l
[[0,7,0],[0,7,0],[0,7,0],[0,7,0]]
```

- Each row in `l` points to same list `zerolist`

NPTEL

Summary

- `map` and `filter` are useful functions to manipulate lists
- List comprehension provides a useful notation for combining `map` and `filter`