6.12 Lambdas

lambda\_expr ::= “lambda” [parameter\_list]: expression

lambda\_expr\_nocond ::= “lambda” [parameter\_list]: expression\_nocond

Lambda expressions (sometimes called lambda forms) are used to create anonymous functions. The expression

lambda arguments: expression yields a function object. The unnamed object behaves like a function

object defined with

def <lambda>(arguments):

return expression

See section Function definitions for the syntax of parameter lists. Note that functions created with lambda expressions

cannot contain statements or annotations.

一組範例:

Python提供了一個簡易的function define：lambda，用完即丟，不著痕跡。讓你實作出很簡單的function(只處理一個運算式)。

lambda param1, param2, ... : expression

*#其實就等於*

def fun( param1, param2, ... ) :

return expression

其中的expression不能放assignment，也就是這一行指令不能放=等號。因為，它就這麼簡單，別把它搞複雜化嘛~  
  
Ex:

def func(x, y, z):

return x + y + z

*#>>> func(1, 2, 3)*

*#>>> 6*

func2 = lambda x,y,z : x+y+z

*#>>> func2(1, 2, 3)*

*#>>> 6*

*#也可以應用在map上*

my\_list = [1, 2, 3]

map( lambda i: i \* i, my\_list )

*#>>> (1, 4, 9)*

lambda在某些方面而言確實是很好用，但是也不能濫用，否則可能造成程式的可讀性降低。