CS 314 Principles of Programming Languages

Guang Wang
Department of Computer Science
Rutgers University

Python lambda (Anonymous Functions)

In Python, anonymous function means that a function is without a name. As we already know that *def* keyword is used to define the normal functions and the *lambda* keyword is used to create anonymous functions.

It has the following syntax:

lambda argument: expression

lambda

- This function can have any number of arguments but only one expression, which is evaluated and returned.
- One is free to use lambda functions wherever function objects are required.
- It has various uses in particular fields of programming besides other types of expressions in functions.

Defined function vs. lambda function

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

f = lambda y: y*y
print(f(5))

print(square(5):)
```

Using Lambda : Lambda definition does not include a “return” statement, it always contains an expression which is returned. We can also put a lambda definition anywhere a function is expected, and we don’t have to assign it to a variable at all. This is the simplicity of lambda functions.

Built-in functions

- filter()

```
List1 = [6, 20, 23, 33, 45, 32]
List2 = list(filter (lambda x: (x%2 !=0), List1))
print(List2)
```

Built-in functions

- map()

```
List1 = [6, 20, 23, 33, 45, 32]
List2 = list(map (lambda x: x*2, List1))
print(List2)
```

Built-in functions

- reduce()

```
from functools import reduce
List1 = [6, 20, 23, 33, 45, 32]
sum = reduce ((lambda x, y: x+y), List1)
print(sum)
```

Process: (((((6 +20)+23)+33)+45)+32)

Thanks!!!