Functional Programming Tools

- Python is not a functional programming language but it does incorporate some of its concepts alongside other programming paradigms.
- Python can support functional programming through some handy functional tools
- We’ll take a look at some of functional tools provided by Python.

Lambda

Lambda Functions

- Lambda functions: a small anonymous function that appear in many functional languages

lambda *arguments* : *expression*

  - Use the keyword *lambda* instead of *def*.
  - Can be used wherever function objects are used.
  - **Restricted to one expression**.
  - Typically used with functional programming tools, like g=g(x).

```
def f(x):
    return x**2
print(f(8))
64

g = lambda x:x**2
print(g(8))
64
```

Lambda Functions

- Lambda functions:

```
lambda arguments : expression
```

A lambda function can take any number of arguments, but can only have one expression

```
x = lambda a, b : a * b
print(x(5, 6))
```

Lambda Functions

- Lambda functions

```
def myfunc(n):

    return lambda a:a*n

mydoubler = myfunc(2)

print(mydoubler(11))
```

$\longrightarrow$ mydoubler = lambda a:a*2 $\longrightarrow$ mydoubler(a)

What is results? 22