Python Lambda

What are lambda functions in Python?

In Python, an anonymous function is a function that is defined without a name.

While normal functions are defined using the def keyword in Python, anonymous functions are defined using the lambda keyword.

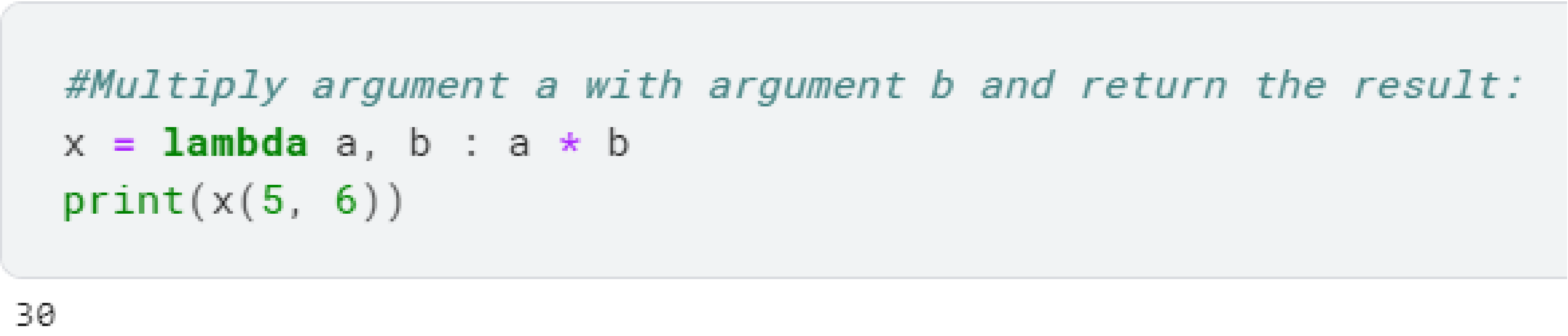
Hence, anonymous functions are also called lambda functions.

How to use lambda Functions in Python?

- A lambda function in python has the following syntax.

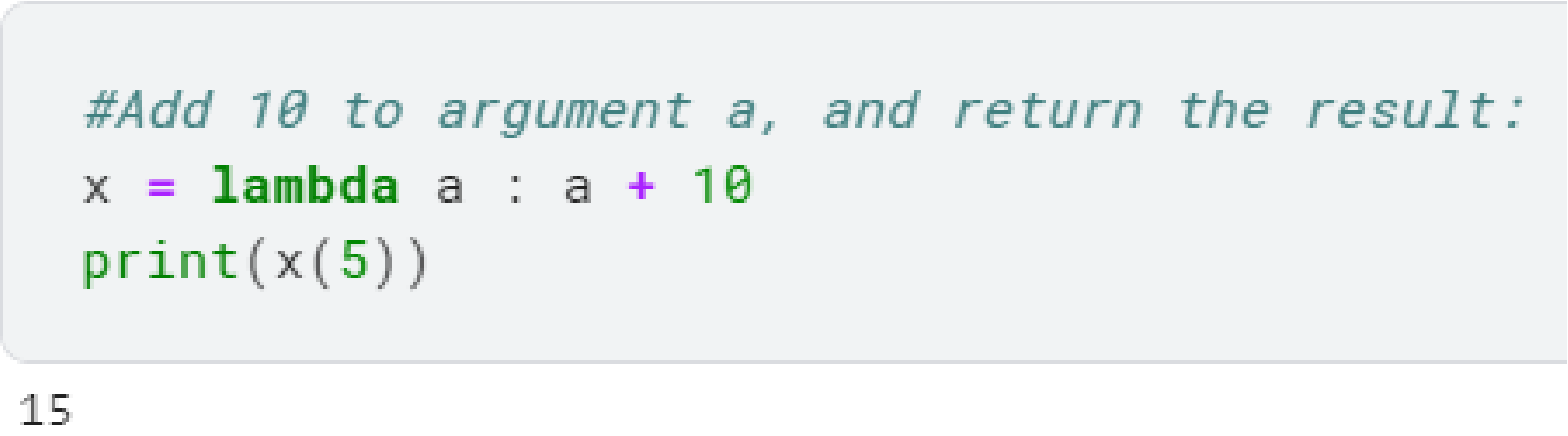
lambda arguments: expression

- Lambda functions can have any number of arguments but only one expression. The expression is evaluated and returned. Lambda functions can be used wherever function objects are required.



```
#Multiply argument a with argument b and return the result:
x = lambda a, b : a * b
print(x(5, 6))
```

```
30
```



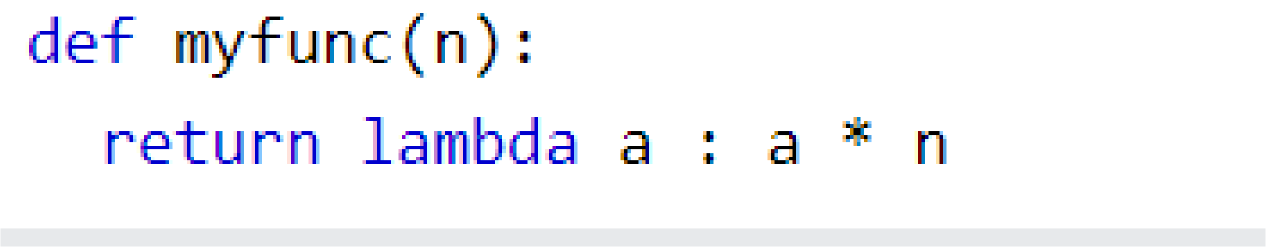
```
#Add 10 to argument a, and return the result:
x = lambda a : a + 10
print(x(5))
```

```
15
```

Examples

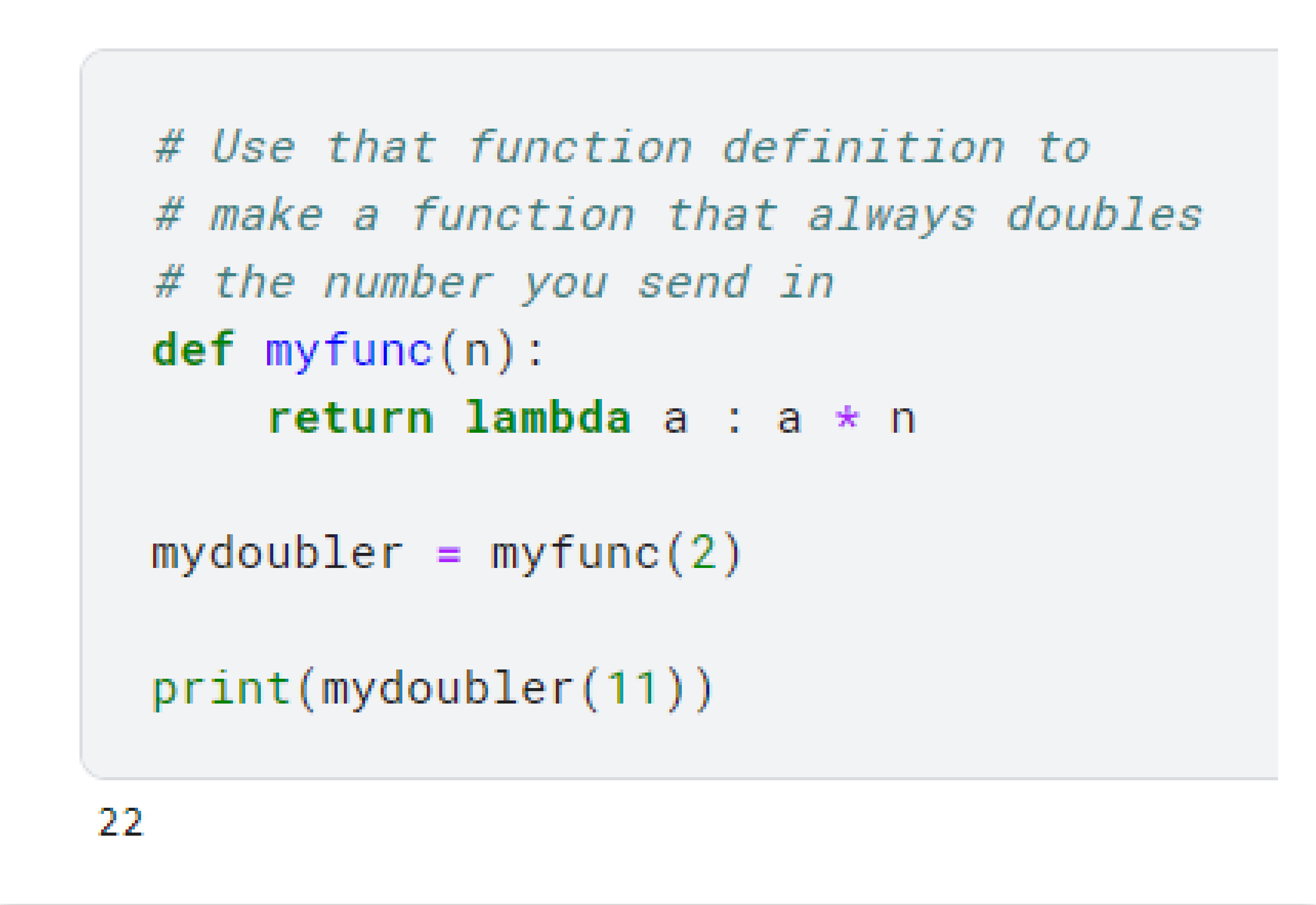
Why Use Lambda Functions?

- The power of lambda is better shown when you use them as an anonymous function inside another function.
- Say you have a function definition that takes one argument, and that argument will be multiplied with an unknown number:



```
def myfunc(n):
  return lambda a : a * n
```

Example



```python
# Use that function definition to
# make a function that always doubles
# the number you send in
def myfunc(n):
    return lambda a : a * n

mydoubler = myfunc(2)

print(mydoubler(11))
```

```
22
```