

# Lecture12

October 10, 2024

```
[1]: def apply_function(func, value):  
      return func(value)  
  
      def square(x):  
          return x * x  
  
      print(apply_function(square, 5))
```

25

```
[2]: name = "Alice"  
      new_name = name.upper()  
  
      print(name)  
      print(new_name)
```

Alice  
ALICE

```
[4]: def greet(name):  
      return f"Hello, {name}!"  
  
      say_hello = greet  
      print(say_hello("Alice"))
```

Hello, Alice!

```
[5]: add = lambda x, y: x + y  
      print(add(3, 5))  
  
      numbers = [1, 2, 3, 4, 5]  
      squared_numbers = list(map(lambda x: x * x, numbers))  
      print(squared_numbers)
```

8

[1, 4, 9, 16, 25]

```
[6]: numbers = [1, 2, 3, 4, 5]  
      even_numbers = list(filter(lambda x: x % 2 == 0, numbers))  
      print(even_numbers)
```

[2, 4]

```
[7]: from functools import reduce

numbers = [1, 2, 3, 4, 5]
sum_of_numbers = reduce(lambda x, y: x + y, numbers)
print(sum_of_numbers)
```

15

```
[10]: from functools import reduce
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
squared_numbers = map(lambda x: x ** 2, numbers)
even_squares_numbers = filter(lambda x: x % 2 == 0, squared_numbers)

sum_of_even_squared_numbers = reduce(lambda x, y: x + y, even_squares_numbers)
print(sum_of_even_squared_numbers)
```

220

```
[16]: def apply_twice(func, value):
        return func(func(value))

def increment(x):
    return x + 1

print(apply_twice(increment, 5))
```

7

```
[17]: def create_multiplier(n):
        return lambda x: x * n

double = create_multiplier(2)
triple = create_multiplier(3)

print(double(5))
print(triple(5))
```

10

15