

# PyStart #26 Funkcja jako argument, lambda!

## O co chodzi?

- Co może być argumentem funkcji?
- Przekazywanie "sposobu"- "know-how" do funkcji



# PyStart #26 Funkcja jako argument, lambda!

## O co chodzi?

Callable[[argument\_type], return\_type] **tylko 3.9**

```
1  from collections.abc import Callable
2
3
4  #Callable[[argument_type], return_type]
5
6  def do_something(data: list, f: Callable[[list], int]):
7      return f(data)
8
9
10 print(do_something([1, 2, 3], sum))
11 print(do_something([4, 5, 6], len))
12
```



# PyStart #26 Funkcja jako argument, lambda!

## Czym jest lambda?

- Krótkie funkcje, jedno liniowa
- Służą do przekazywania funkcji jako argument.



# PyStart #26 Funkcja jako argument, lambda!

## Czym jest lambda?



```
1  power = lambda x: x**2
2
3  print(power(10))
```

więcej argumentów:

```
1  sum_numbers = lambda a, b: a + b
```

# PyStart #26 Funkcja jako argument, lambda!

## Trzy przydatne funkcje

- **Filter** - usuwanie elementów
- **Map** - zmienianie elementów
- **Reduce** - redukcja :-)

**function\_name**(callback, iterable)

**callback** - funkcja do zastosowania na elementach

**iterable** - lista, słownik



# PyStart #26 Funkcja jako argument, lambda!

## Przykłady funkcji - map



```
1 numbers = [1, 5, 10, 9, 7]
2
3 doubles = [n + n for n in numbers]
4
5 doubles = map(lambda n: n + n, numbers)
6 |
```

# PyStart #26 Funkcja jako argument, lambda!


## Przykłady funkcji - filter



```
1 numbers = [1, 5, 10, 9, 7]
2
3 filtered = [n for n in numbers if n % 2 == 0]
4
5 filtered = filter(lambda n: n % 2 == 0, numbers)
6
```

# PyStart #26 Funkcja jako argument, lambda!

## Przykłady funkcji - reduce



```
1  from functools import reduce
2
3  numbers = [
4      (1, 2),
5      (3, 4),
6      (5, 6)
7  ]
8
9  total = sum([a * b for a, b in numbers])
10 print(total)
11
12 total = reduce(lambda sum, a: sum + a[0] * a[1], numbers, 0)
13 print(total)
```



# PyStart #26 Funkcja jako argument, lambda!

## Przykłady funkcji - sort

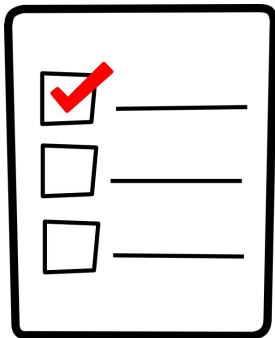


```
1 names = ['Leon', 'Barbara', 'Czesław', 'Fryderyk']
2 by_alphabet = sorted(names)
3 by_last_letter = sorted(names, key=lambda x: x[-1])
4 by_length = sorted(names, key=len)
```

```
['Barbara', 'Czesław', 'Fryderyk', 'Leon']
['Barbara', 'Fryderyk', 'Leon', 'Czesław']
['Leon', 'Barbara', 'Czesław', 'Fryderyk']
```

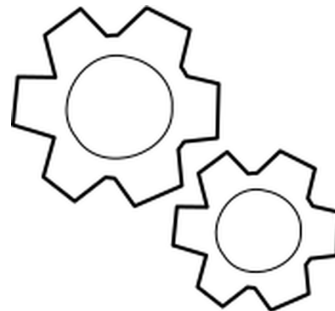
# PyStart #26 Funkcja jako argument, lambda!

Co robić?



List comprehension

vs



funkcje + lambda

# PyStart #26 Funkcja jako argument, lambda!

## Zadania dla nabrania wprawy

!!! PAMIĘTAJ O TYM BY ROZPOCZĄĆ OD TESTÓW !!!

Korzystając z funkcji poznanych w tym rozdziale

## 26.1

1. Policz pierwiastek kwadratowy każdego elementu listy, a następnie przefiltruj tylko te elementy których pierwiastek jest liczbą parzystą. np. [16, 36, 25, 49]
2. Sformatuj listę osób tak, aby imiona i nazwiska były zapisane z wielkiej litery. **Podpowiedź: możesz zadeklarować dodatkowe funkcje.**  
people = ['zofia nowak', 'kryStyna kowalska', 'michał lewandowski'],  
a następnie posortuj te osoby według pierwszej litery nazwiska.

