



Lambda

Stefan Holmberg, Systemmentor AB

Funktioner

- Kan vi väl redan #3??
 - Funktion = First-Class Object
 - ***can be passed around and used as arguments***
 - precis som string,int,object etc etc

En funktion utan () är ett objekt. Först när man lägger till () körs den

Ex

```
def CalculateSalary(): # SRP = endast detta
    # på riktigt en beräkning
    return 100

def CalculateWhatever(): # SRP = endast detta
    # på riktigt en beräkning
    print("Starting")
    x = 200
    print("Ending")
    return x

def print_decorator( func ):
    print("Starting")
    func()
    print("Ending")

salary = print_decorator(CalculateSalary)
whatever = print_decorator(CalculateWhatever)
```

Use case?

Återanvända en "algorithm" med olika specifika saker insprängda i den

Ex säg att vi har en algorithm (funktion) som läser (alla) rader i (alla) filer i ett bibliotek

```
import os
def ScanFiles(dir):
    for fileName in os.listdir(dir):
        if fileName.endswith(".txt"):
            with open(fileName,"r") as file:
                for line in file:
                    l = line.replace("\n", "")
                    #Do something
                    pass
```

Use case?

Kanske vill vi använda denna algoritm från många ställen för att hantera olika typer av filer... tex

```
def ScanFiles2(dir, func):
    for fileName in os.listdir(dir):
        if fileName.endswith(".txt"):
            with open(fileName, "r") as file:
                for line in file:
                    l = line.replace("\n", "")
                    func(line)
```

```
def LookForBadIps(line):
    badIps = ["1.2.3.4", "3.4.5.6"]
    for x in badIps:
        if x in line:
            print(f"Bad ip {x} found")
    return
```

```
def LookForDeposits(line):
    if "deposit" in line:
        print(f"Deposit {x}")
        with open("deposits.txt") as f:
            f.write(line + "\n")
```

```
ScanFiles2(".\\logfiles\\", LookForBadIps)
ScanFiles2(".\\transactions\\", LookForDeposits)
```

Så funkar sort!

```
from dataclasses import dataclass
```

```
@dataclass  
class Player:  
    Name:str  
    Goals: int  
    Assists: int
```

```
lista = [Player("Stefan",12,22), Player("Mats",11,99), Player("Foppa",11,77)]
```

```
def sortFunc(p):  
    return p.Goals + p.Assists
```

```
lista.sort(key=sortFunc, reverse=True)  
print(lista)
```

Jobbigt med att skapa en funktion

```
from dataclasses import dataclass
```

```
@dataclass  
class Player:  
    Name: str  
    Goals: int  
    Assists: int
```

```
lista = [Player("Stefan",12,22), Player("Mats",11,99), Player("Foppa",11,77)]
```

```
lista.sort(key=lambda p: p.Goals + p.Assists, reverse=True)
```

```
print(lista)
```

Lambda?

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.

Usecaset som inte är ett usecase

Map och Filter jobbar med lambdas

<https://www.programiz.com/python-programming/anonymous-function>

Men det är best practice att göra LIST COMPREHENSIONS istf att använda just map och filter