

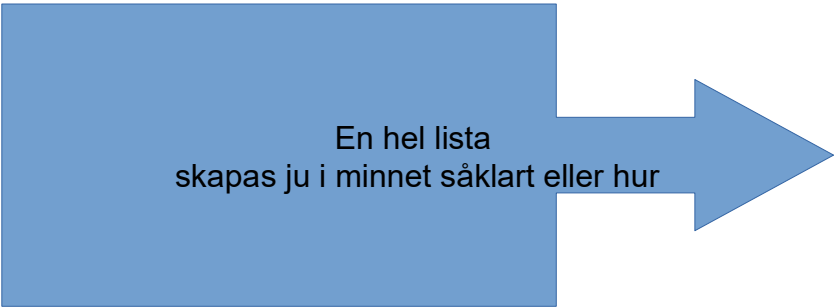
Generators

Stefan Holmberg, Systementor AB

Listor – the problem

```
@dataclass
class Person:
    Namn:str
    PersonNummer: str
```

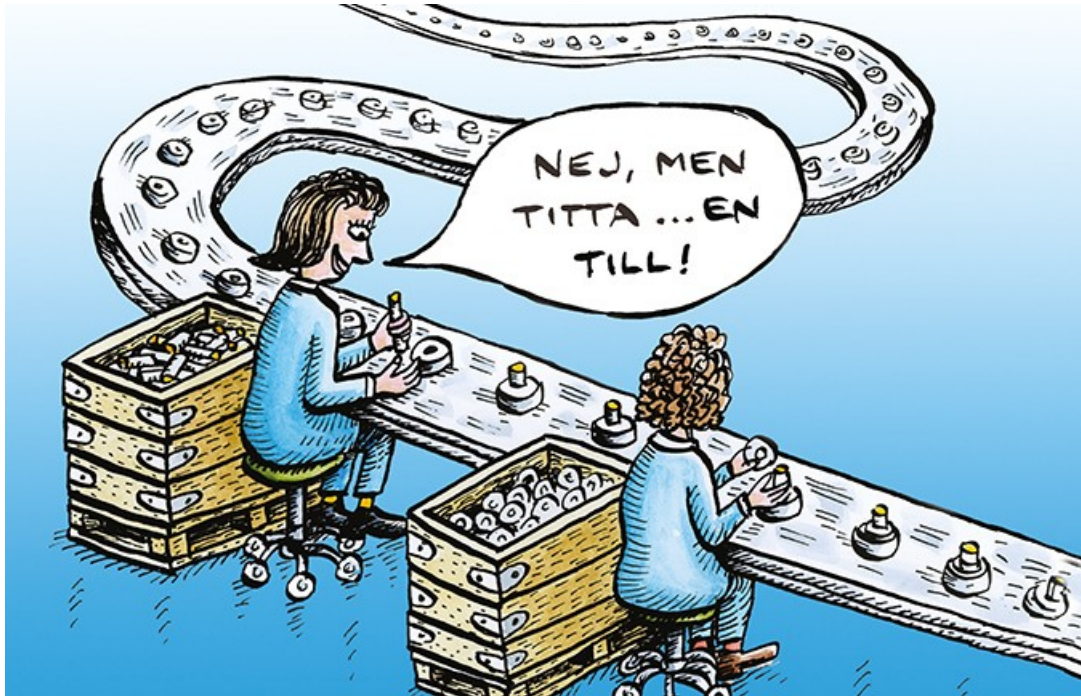
```
def getAllPersons() -> list[Person]:
    # vi läser från databas
    # men vi fejkar här med en loop
    result = []
    for x in range(0,1000000):
        namn = f"Person {x}"
        personNummer = str(x).zfill(10)
        result.append(Person(namn,personNummer))
    return result
```



En hel lista
skapas ju i minnet såklart eller hur

```
for x in getAllPersons():
    print(f"Sending mail to {x.PersonNummer}")
```

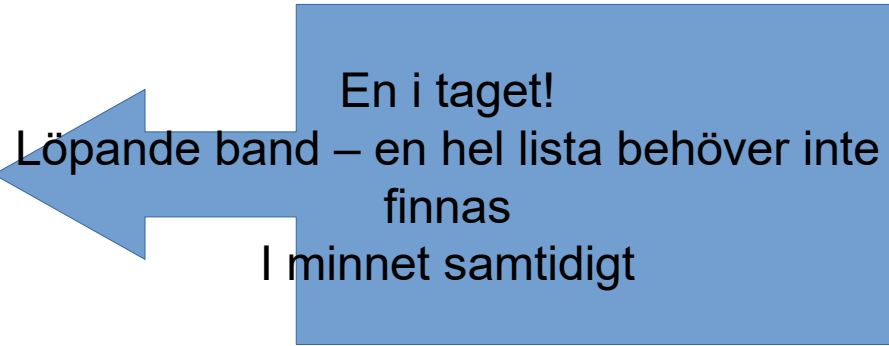
Generators = Löpande band!



Generator

```
def getAllPersons2() -> Iterator[Person]:  
    # vi läser från databas  
    # men vi fejkar här med en loop  
    for x in range(0,1000000):  
        namn = f"Person {x}"  
        personNummer = str(x).zfill(10)  
        yield Person(namn,personNummer)
```

```
for x in getAllPersons2():  
    print(f"Sending mail to {x.PersonNummer}")
```



En i taget!
Löpande band – en hel lista behöver inte
finnas
I minnet samtidigt

Listor – plus och minus

```
# Lista = [] - alla saker finns i minnet
#           möjlighet att läsa framåt + bakåt +
#           random access 18, 99
massa = [CanOfSoup(True,False,False) for x in range(0,100)]
print(massa)
for x in massa:
    print(x)

for index in range(len(massa)-1,0,-1):
    print(massa[index])
print(massa[45])
```

Generator – en i taget

```
# CREATE = Generator
# YIELD = puttar ut EN på löpande bandet
# ger kontrollen tillbaka till den som anropade funktionen
# i nästa varv i loopen kommer Create komma ihåg var den slutade
# och fortsätta därifrån
```

```
def Create(n): # Jag var på yield raden och x = 1
    for x in range(0,n): # x = 1
        sleep(3)
        yield CanOfSoup(True,False,False)
```

```
def PutOnLabel(canOfSoup):
```

```
    canOfSoup.Label = True
```

```
def PourSoup(canOfSoup):
```

```
    canOfSoup.Soup = True
```

```
#
```

```
#VARJE VARV I LOOPEN
# drar fram en till YIELD från Create
#
for x in Create(100000000):
    PutOnLabel(x)
    PourSoup(x)
    print(x)
```

```
for x in range(0,100000000):
    x = CanOfSoup(True,False,False)
    PutOnLabel(x)
    PourSoup(x)
    print(x)
```