

# GruProg DD1331

## Övning 5

Anna Karlhede

karlhed@kth.se

Repo: <https://github.com/TypAnna/GruProgDD1331>

# Idag

- Repetition/genomgång
  - Klasser, arv
  - Mer om funktioner
  - P-uppgiften
- Uppgifter
  - minnesallokering
  - klasser och gui

# Repetition



# Klass och arv

Vi kan bygga ut redan existerande (antingen inbyggda eller egenskapade) klasser genom att låta en klass ärva från en annan.

Låt säga att vi har en klass `Human` med massa inbyggda attribut (properties) såsom `age`, `length`, `shoeSize`, `interests` och metoder såsom `yell`, `washDishes`, `cry`.

Vi kan då skapa nya klasser, så som `class Athlete(Human)`, `class Professor(Human)` som ärver från `Human`-klassen. På instanser av `Athlete` och `Professor` kommer vi kunna använda `Human`-metoderna. i.e. vi kan skriva

```
anAthlete.yell("jaaa")  
anProfessor.yell("läs boken :-")
```

Utöver det kan vi definiera metoder som är specifika för `Athlete` resp. `Professor`.

Leder till mindre kodupprepning och fin struktur samt att vi kan utnyttja redan existerande klasser (behöver alltså inte uppfinna hjulet på nytt).

# Funktioner - named arguments

```
def testa(a, b):  
    print("a", a)  
    print("b", b)
```

```
#antingen specificerar vi vilken parameter som ska vara vad mha  
ordningen  
hej("okej", "wioho")
```

```
#...eller genom att explicit säga vilken parameter som är vilken  
hej(b="wioho", a="okej")
```

# Funktioner - anonyma funktioner

Anonyma funktioner är funktioner utan namn.

Vanliga funktioner definierar vi med nyckelordet `def`.

Anonyma funktioner definierar vi med nyckelordet `lambda`.

En anonym funktion kan ta hur många parametrar (argument) som helst, men består alltid enbart av ett expression och returnerar resultatet av detsamma. Alltså är det väldigt korta funktioner.

Smidigt när vi har att göra med korta funktioner som finns en kort stund.

# P-uppgiften

Läs *noga* igenom hela sidan om [P-uppgiften](#)

Din kod ska inte bara fungera (i.e. lösa uppgiften) utan vara välstrukturerad och lättförståelig.  
Obs: detta är en KRAV för att bli godkänd.

# Uppgifter





# Uppgift 1- Klass och minne

Rita en minnesbild med låd- och pildiagram för hur det ser ut då körningen når den kommenterade raden.

```
class Item:
    """A singly linked list."""
    def __init__(self, value=None):
        self.value = value
        self.next = None

def main():
    first = Item()
    first.next = Item()
    first.next.next = Item()
    first.next = first.next.next
    #rita hur minnet ser ut här

main()
```

# Uppgift 2- Klass och minne

Vad skriver följande Python3-program ut?

```
def mystery(v, n):  
    n = n + 1  
    v[1] = n
```

```
def main():  
    v = [1, 2]  
    n = 4  
    mystery(v, n)  
    print(v, n)
```

```
main()
```

# GUI-uppgifter

Se kurshemsidan för att se GUI-uppgifterna. De uppgifterna vi gick igenom ligger uppe på min GitHub med kommentarer.