

Skolan för Datavetenskap och kommunikation

DD1319

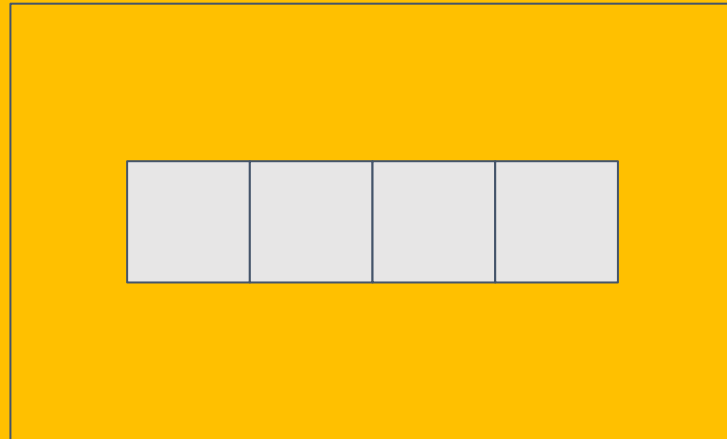
Programmeringsteknik

Föreläsning 9

idag

- Litet exempel med lista som attribut
- `random.seed()`
- Python-visualiseraren
- Arv

lista som attribut



klassen Lotteri

```
class Lotteri:
```

```
    def __init__(self):  
        self.vinst = "en diskmaskin"  
        self.vinstlista = [4, 50, 88, 96]
```

main

```
def main():  
    lotteri = Lotteri()  
    lott = randint(1,100)  
    print("Din lott har nummer ", lott)  
    if lott in lotteri.vinstlista:  
        print("Du vann", lotteri.vinst)  
    else:  
        print("nitlott")
```

hela programmet

```
from random import randint
```

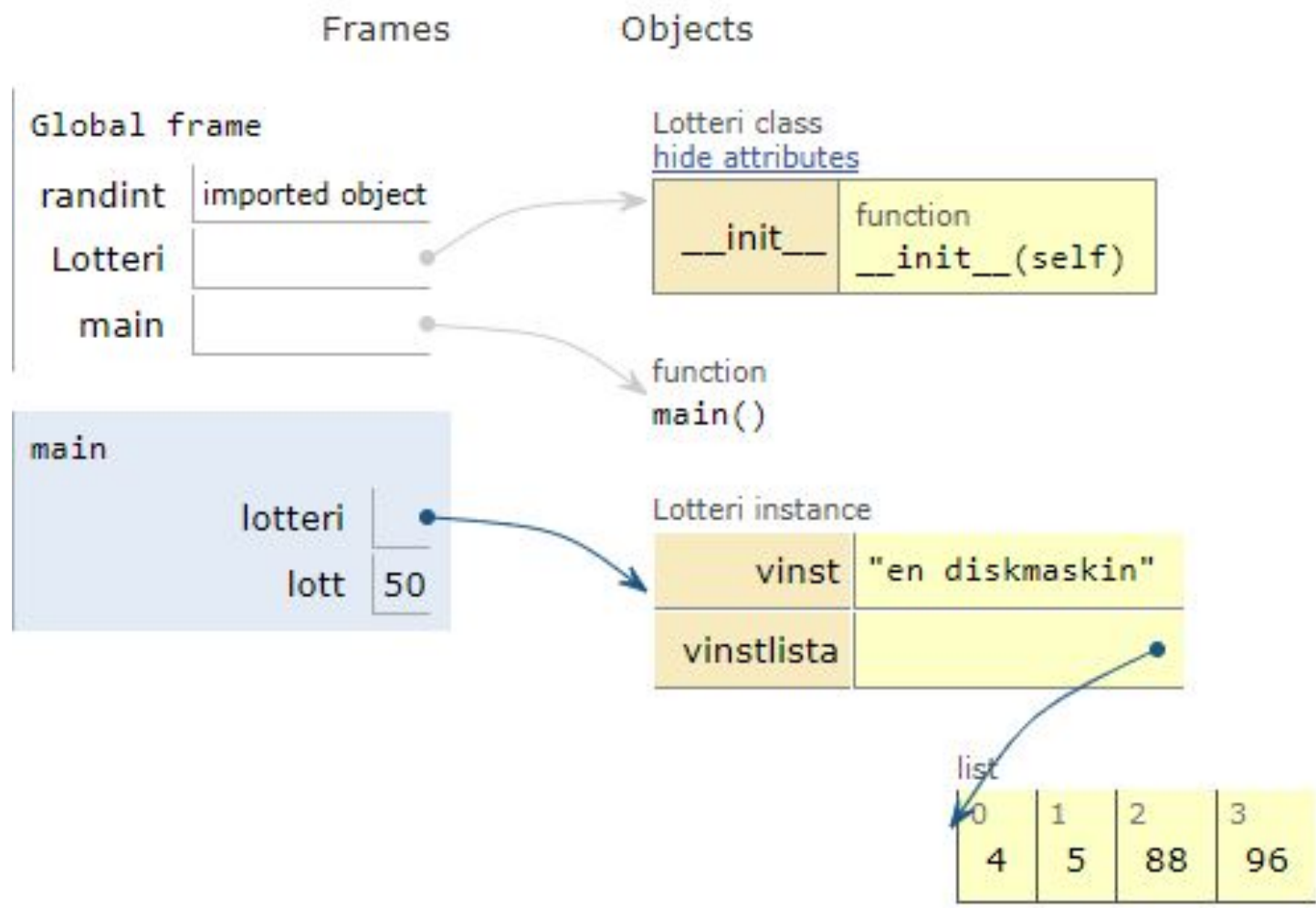
```
class Lotteri:
```

```
    def __init__(self):  
        self.vinst = "en diskmaskin"  
        self.vinstlista = [4, 50, 88, 96]
```

```
def main():  
    lotteri = Lotteri()  
    lott = randint(1, 100)  
    print("Din lott har nummer ", lott)  
    if lott in lotteri.vinstlista:  
        print("Du vann", lotteri.vinst)  
    else:  
        print("nitlott")
```

```
main()
```

pythontutor.com/visualize.html



pythontutor: begränsningar

- Filhantering saknas

hela programmet

```
from random import randint, seed
```

```
class Lotteri:
```

```
    def __init__(self):  
        self.vinst = "en diskmaskin"  
        self.vinstlista = [4, 50, 88, 96]
```

```
def main():  
    seed(0)  
    lotteri = Lotteri()  
    lott = randint(1, 100)  
    print("Din lott har nummer ", lott)  
    if lott in lotteri.vinstlista:  
        print("Du vann", lotteri.vinst)  
    else:  
        print("nitlott")
```

```
main()
```

seed

- seed är en metod i random
- Ett anrop av seed, t ex
 seed(8)
ger ett fixt startvärde för slumpserien
- Användbart vid testning!

Vill du veta mer om pseudoslumftal? Läs i [Wikipedia](#)

Fråga

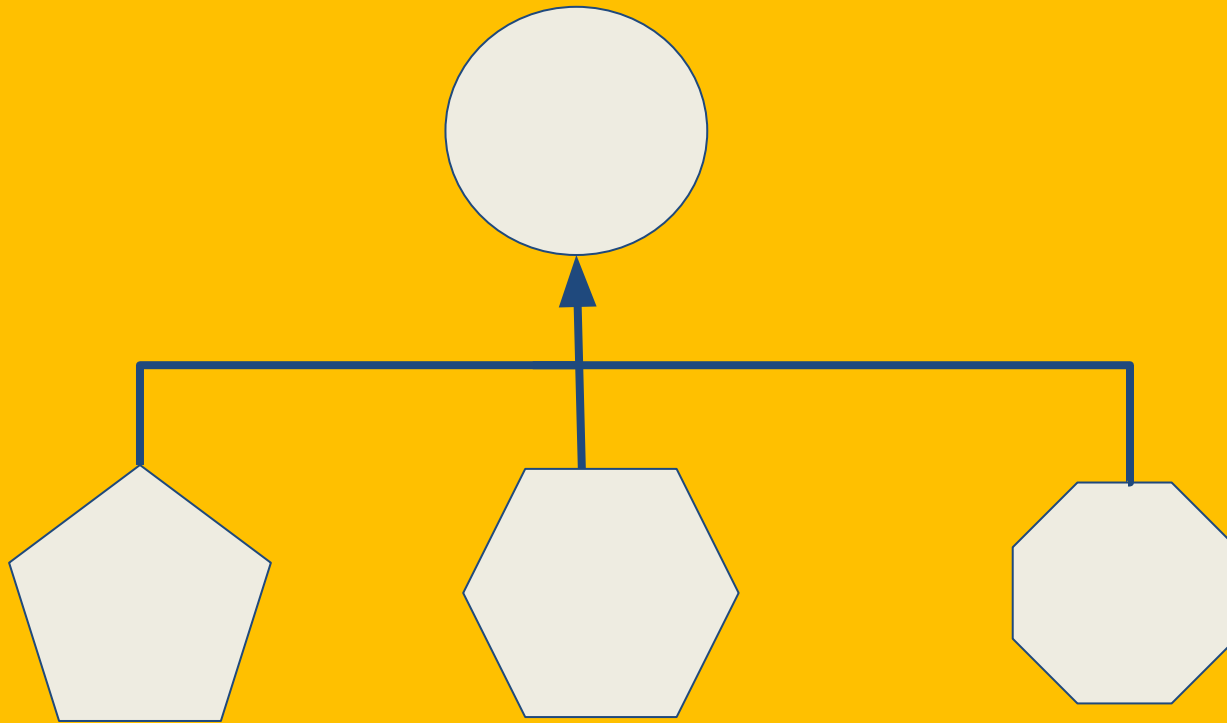
Nu kan ju samma lottnummer slumpas fram flera gånger. Hur ser man till att varje nummer bara kan slumpas fram en gång?

Vi ser till att unika lottnummer slumpas

```
def main():
    redan_lottad = []
    lotteri = Lotteri()
    seed(2023)
    for i in range(4):
        lott = randint(1,100)
        while lott in redan_lottad:
            lott = randint(1,100)
        redan_lottad.append(lott)
        print("Din lott har nummer ", lott)
        if lott in lotteri.vinstlista:
            print("Du vann", lotteri.vinst)
        else:
            print("nitlott")
```

```
main()
```

arv



arv

- Kap 11 i boken
- superklass
- subclass
- klassen object

arv = inheritance

kodåtervinning

- Samma satser om igen?
 - *Skriv en slinga eller en funktion!*
- Samma funktioner i nytt program?
 - *Skriv en modul och importera den!*
- Samma data som skickas in i alla funktionerna?
 - *Skriv en klass!*
- Flera klasser med liknande attribut och metoder?
 - *Skriv en superklass och låt klasserna ärva från den!*

Exempel: Bok

- *Bok* är den mest generella figuren - den får bli superklass
- *Ljudbok* är en sorts *Bok* - vi låter den vara subklass till *Bok*

Klassen Bok

Attribut	Metoder
titel författare	__init__ __str__

```
class Bok:
```

```
    def __init__(self, titel, författare):  
        self.titel = titel  
        self.författare = författare
```

```
    def __str__(self):  
        return "'" + self.titel + "' skriven av " + self.författare
```



Klassen Ljudbok

Attribut	Metoder
titel författare uppläsare	__init__ __str__ ange_uppläsare

```
class Ljudbok(Bok):
```

```
    def ange_uppläsare(self, namn):  
        self.uppläsare = namn
```



```
def main():  
    bok1 = Bok("Blackout", "C. Willis")  
    print(bok1)  
  
    bok2 = Ljudbok("Musicophilia", "O. Sacks")  
    print(bok2)  
  
    bok2.ange_uppläsare("John Lee")  
    print(bok2.uppläsare)
```

hela programmet

```
class Bok:

    def __init__(self, titel, författare):
        self.titel = titel
        self.författare = författare

    def __str__(self):
        return ""+self.titel + " skriven av " + self.författare

class Ljudbok(Bok):

    def ange_uppläsare(self, namn):
        self.uppläsare = namn

def main():
    bok1 = Bok("Blackout", "C. Willis")
    print(bok1)

    bok2 = Ljudbok("Musicophilia", "O. Sacks")
    print(bok2)

    bok2.ange_uppläsare("John Lee")
    print(bok2.uppläsare)

main()
```

Arv

```
class Subklass(Superklass):
```

Subklassen ärver alla

attribut och metoder

från klassen Superklass.

object

När du skriver

```
class MinKlass:
```

översätter Python det till

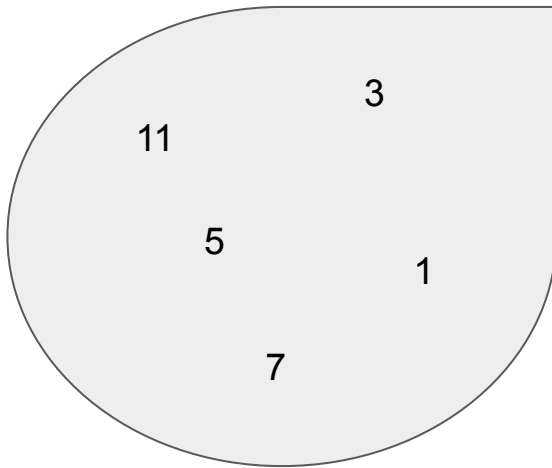
```
class MinKlass(object):
```

Det betyder att MinKlass ska ärva från *object*
object är superklass till *alla* klasser i Python!

inför Labb 6: set

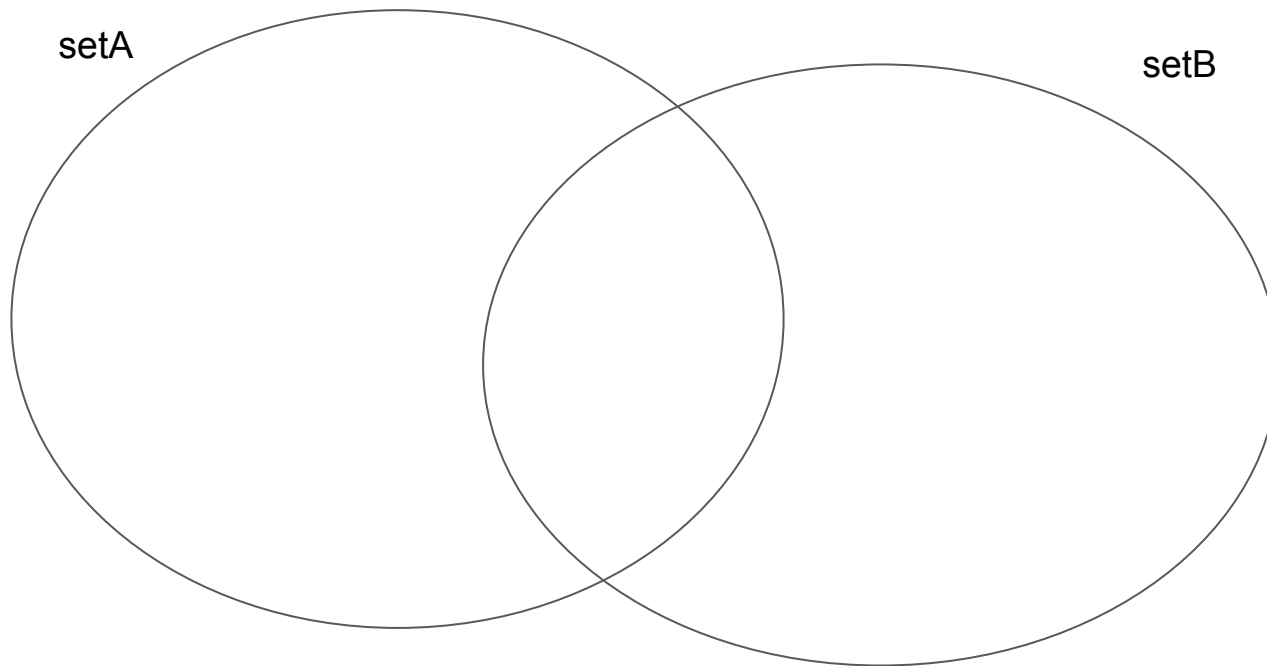
docs.python.org/3/library/stdtypes.html#set

set



set - union

Vad ingår i unionen?



set - snitt (intersection)

Vad ingår i snittet?

