



Ny datatyp: objekt

- Kap 8 i Dawson
- Klasser
- Objekt
- Attribut och metoder

Objekt

- Objekt är en datatyp som kan innehålla både data och funktioner.
- Data som tillhör objektet kallas *attribut*.
- En funktion som hör till objektet kallas *metod*.



Du har redan använt objekt i Python!

Klass	Attribut	Metodanrop
String bok="1q84"	Alla tecken i strängen	s=bok.upper()
List regn = [19, 5, 12]	Alla element i listan	regn.sort()

Husdjur

Klass	Attribut	Metoder
Husdjur	namn glad hunger	leka() banna() mata(bullar) visaStatus()



Klasser

- Den del av programmet där man definierar attribut och metoder kallas en *klass*.
- Klassen fungerar som mall för objekt
- I klassen beskrivs attribut och metoder.
- Alla metoder har parametern `self` som används för att komma åt de egna attributen.

```
class Husdjur(object):  
    """ Ett virtuellt husdjur """  
    def __init__(self, djurnamn):  
        self.namn = djurnamn  
        self.glad = 0  
        self.hunger = 0  
    def banna(self):  
        """Ger husdjuret bannor."""  
        print "- Fy på dig", self.namn, "!"  
        self.glad -= 3  
    def mata(self, mat):  
        """Ger husdjuret mat."""  
        for i in range(mat):  
            print "GLUFS"  
            self.hunger -= mat
```

attribut

metoder

Skapa objekt ur klassen

- Satsen `djur = Husdjur("Totoro")` skapar ett objekt
- Konstruktorn `__init__` anropas automatiskt
- Attributen ges värden
- En referens till objektet returneras

Metodanrop

- Metoder anropas alltid via objektet:
`djur.leka()`
`djur.banna()`
`djur.mata(bullar)`
`djur.visaStatus()`
- I övrigt fungerar metoder som vilken funktion som helst (kan ha parametrar och returvärden)

Klass, instans, self

- En *klass* är en generell beskrivning av vad objekten ska innehålla/kunna göra.
- Ett *objekt* är en *instans* av en klass. Variabeln `djur` refererar till ett objekt.
- I klassdefinitionen använder man *self* för att hänvisa till det anropande objektet.
`djur.banna()` `banna(djur)`

```
def visaStatus(self):  
    """Visar husdjurets namn och hur det mår."""  
    print "\n", self.namn, "är",  
    if self.glad > 0 and self.hunger < 3:  
        print "glad: (^_^)"  
    else:  
        print "ledsen: (T_T)"
```

```

def __str__(self):
    """Returnerar en sträng som beskriver husdjuret"""
    beskrivning = self.namnet + " är "
    if self.glad > 5:
        beskrivning += "glad: (^_^)"
    else:
        beskrivning += "ledsen: (T_T)"
    if self.hunger > 3:
        beskrivning += " och hungrig!"
    else:
        beskrivning += " och mätt."
    return beskrivning

```

```

def main():
    djurnamn = raw_input("Vad vill du kalla djuret? ")
    djur = Husdjur(djurnamn)
    djur.visaStatus()
    svar = raw_input(" Vill du \n banna \n mata \n leka
med \n ditt husdjur? " )
    while svar:
        if svar[0]=="m":
            bullar = input("Hur många bullar? ")
            djur.mata(bullar)
        elif svar[0]=="b":
            djur.banna()
        elif svar[0]=="l":
            djur.leka()
        else:
            print "Hursa? "
            djur.visaStatus()
            svar = raw_input(" Vill du \n banna \n mata \n
leka med \n ditt husdjur? " )
    djur.avsked()

```