



### 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*.

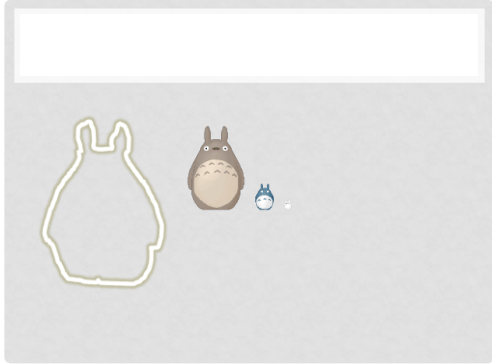
*namn*: "Totoro"  
*glad*: 0  
*hunger*: 5

### DU HAR REDAN ANVÄNT OBJEKT I PYTHON!

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

### EXEMPEL: HUSDJUR

Klass	Attribut	Metoder
Husdjur	<i>namn</i> <i>glad</i> <i>hunger</i>	<code>leka()</code> <code>banna()</code> <code>mata(ekollon)</code> <code>visaStatus()</code>



## 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 nå 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()` betyder `banna(djur)`

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

```

def __str__(self):
    """Returnerar en sträng
    som visar husdjurets tillstånd"""
    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 = input("Vad vill du kalla djuret? ")
    djur = Husdjur(djurnamn)
    djur.visaStatus()
    svar = input(" Vill du \n banna \n mata \n leka med \n
ditt husdjur? ")
    while svar:
        if svar[0]=="m":
            bullar = int(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 = input(" Vill du \n banna \n mata \n leka med \n
ditt husdjur? ")
    djur.avsked()

```

## LÄXA 3

Se "Labbar, föreläsningar och övningar" på kurswebbsidan.

- I labb 4 ska du skriva ett program med en egen klass.
- Läs Kap 8 i (om klasser) i kursboken.
- Gå igenom och provkör husdjursprogrammet så att du förstår hur det fungerar.
- Skriv sedan en egen klass (tillsammans med din gruppkompis) med minst fem attribut och metoder.