

DD1311 Programmeringsteknik med PBL

Föreläsning 14

Exempel

```
>>>pris = 1890
>>>extrapris = pris - 0.01
>>>print "Pris %i" % pris
Pris 1890
>>>print "Pris %8i" % pris
Pris      1890
>>>print "REA %.2f" % extrapris
REA 1889.99
```

Snygg utskrift

- Hur skriver man ut en tabell med raka kolumner? Eller ett värde med två decimaler?
- Ange *formatering* i print-satsen!
- Skriv `print a % b` där *a* är en sträng med formateringskoder och *b* är en variabel (eller en tuppel med flera variabler).

Några formateringskoder

%7s	Sträng i 7 positioner	_ _ _ k a t t
%-7s	Sträng i 7 positioner, vänsterjusterat	k a t t _ _ _
%4d	Heltal i 4 positioner	_ _ 1 7
%5.2f	Flyttal i 5 positioner med 2 decimaler	_ 3 . 1 4

Tabell – exempel

```
#Rubrik
print "%-55s %5s %10s" % \
    ("Film", "Betyg", "Visningar")
#Tabellrader
for film in listan:
    print "%-55s %5.1f %10d" % \
        (film.namn, film.medelBetyg(), \
         film.visningar)
```



Olika typer av fel



- Felavbrott (Exception) när programmet körs
- Inget händer när man kör programmet
- Massor av text rinner över skärmen
- Programmet gör något annat än det man ville
- Programmet gör rätt för vissa indata, men inte för andra

@staticmethod

- En metod som inte använder objektets attribut kan definieras som en statisk metod. Se hur det ser ut i programmet [filmer.py](#)

Felavbrott

Lär dig tolka felutskrifter!

Traceback (most recent call last):

```
File "filmer.py", line 124, in <module>
    titta(listan)
```

```
File "filmer.py", line 102, in titta
    film.ny_visning(1)
```

```
TypeError: ny_visning() takes exactly 1
argument (2 given)
```

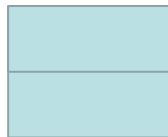
Tolkning

- Sista raden förklarar felet!
`TypeError: ny_visning() takes exactly 1 argument (2 given)`
- Raderna ovanför visar anropskedjan. Läs nerifrån och uppåt! Felet uppstod på rad 102, i funktionen titta som anropades på rad 124.
- Se kurswebbens [FAQ](#): **Felmeddelanden**

Mittiprick-metoden

Anta att programmet hänger sig, men vi vet inte var i programmet det inträffar.

1. Lägg en kontrollutskrift i början och en i slutet. Blev det fel däremellan?
2. Lägg då in en kontrollutskrift mitt i. Om den kommer ut som den ska finns felet i andra halvan, annars i första.
3. Fortsätt tills du hittat felet!



Kontrollutskrifter

- Använd kontrollutskrifter för att hitta var i programmet felet uppstår.
- En kontrollutskrift är en vanlig print-sats, till exempel:

```
print "Klar med inläsningen"
```
- Eller stanna upp så här:

```
raw_input("Tryck Enter")
```
- Du kan också skriva ut variabelvärden för att se hur dom ändras under körning.

Testning

- Skriv upp (i en textfil) hur du testat programmet. Missa inte specialfallen.
- Efter att du har fixat ett fel – testa att programmet fortfarande fungerar för *alla* testfallen.



Hjälpmedel (överkurs)

- Är du van att använda en debugger?
Titta på modulen [pdb](#)
- Vill du infoga automatisk testning i programmet?
Titta på modulen [doctest](#)

Nästa vecka

- Felhantering (ingår ofta i C-uppgiften)
- Dokumentation (hur mycket ska man kommentera?)
- Grafiska gränssnitt (GUI)