

## PROGRAMMERINGSTEKNIK

Föreläsning 16



- teckenkodning

- teckenkodning
- formaterad utskrift
- felhantering
- grafiska gränssnitt

## LAGRING AV DATA

Det här handlar om data du använder i ditt program, t ex från en infil, eller inläst med input.

Alla tecken (bokstäver, siffror, skiljetecken) lagras som binära tal (en kod). Exempel:

- A 100001
- B 100010
- C 100011

## TECKENKODNING

Det finns olika sätt att lagra koderna, och detta spelar roll för oss som lever i ett ääö-land!

Du behöver känna till:  
ISO-8859-1 (även kallad Latin-1), vanligast på Windows  
UTF-8, vanligt på UNIX/Ubuntu/Mac

## ÖPPNA FIL

Så här öppnar du en fil för en viss teckenkodning:  
`infil=open("word3.txt", encoding="utf-8")`

## 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).

## 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
```

## NÅGRA FORMATERINGSKODER

%7s	Sträng i 7 positioner	_ _ _ k a t t
%-7s	Sträng i 7 positioner, vänsterjusterat	k a t t _ _ _
%4i	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 %10i" % (film.namn, film.medelBetyg(), film.visningar))
```

## FELHANTERING

```
try:
    tal = int(input("Ge ett heltal: "))
except ValueError:
    print("Det där var inte ett heltal!")
```

## UPPREPA...

```
def envisInläsning(fråga):
    OK = False
    while not OK:
        try:
            tal = int(input(fråga))
        except ValueError:
            print("Inte heltal - försök igen.")
        else:
            OK = True

envisInläsning("Ge ett heltal: ")
```

## GRAFISKT ANVÄNDARGRÄNSSNITT (GUI)

Använd modulen **tkinter**, som har klasser för grafiska *komponenter*.

Button	Menu
Canvas	Menubutton
Checkbutton	Message
Entry	Radiobutton
Frame	Scale
Label	Scrollbar
Listbox	Text

## ENKLASTE

Hämtar alla klasser i modulen Tkinter.

```
from tkinter import *
roten = Tk()
knapp = Button(roten, text="Tryck")
knapp.pack()
roten.mainloop()
```

Tk-konstruktorn - skapar rotfönstret.

Button-konstruktorn - skapar en knapp.

Knappen placeras.

Startar en slinga som väntar på inmatning från användaren.

## KOMPONENTER

- Knappar och annat kallas *komponenter* och är objekt.
- Varje komponent har en konstruktor med många defaultparametrar.
- Anropa bara med det som behövs:  
`knapp = Button(roten, text="Handla")`
- Första parameter ska vara roten

## ÄNDRA ATTRIBUT

- Attributen kan ändras ett i taget:  
`knapp["text"] = "Klart"`
- Med metoden *config* kan man ändra flera attribut åt gången:  
`knapp.config(bg = "lightblue",  
              height = 3, width = 9,  
              font = ('times', 20, 'italic'))`
- Här ändrar vi knappens färg, storlek, och textfont.

## ANROPA FUNKTION MED KNAPPTRYCK!

- Ett attribut som alla komponenter har är *command*
- Där anger man vilken metod/funktion som ska anropas när komponenten används.
- Om vi skriver en funktion *addera()* som ska anropas när någon trycker på knapp så kan vi koppla ihop funktion med knapp så här:  
`knapp["command"] = addera`

```
from tkinter import *
def byttext():
    knapp["text"] = "Aj!"
roten = Tk()
knapp = Button(roten,
               text = "Tryck inte",
               command = byttext)
knapp.pack()
roten.mainloop()
```



## LÄNKAR & EXEMPEL

- ...finns på KTH Social under "GUI med tkinter"