

DD1311 Programmeringsteknik med PBL

Föreläsning 15 & 16

Mer till P-uppgiften:

- Hjälpssidor
- Sökning
- Rekursion
- Felhantering
- Grafiska gränssnitt

Nästa tisdag:

- Mer om grafiska gränssnitt
- Spelprogrammering med Pygame

Nyheter

- Nu har Pontus rest till Japan! Per-Anders Staav tar över grupp 10.
- P-redovisningar kl 13-15 (för dom som anmält sig).

Dokumentation

- På <http://www.python.org/doc/> finns länkar till dokumentationen:
- **Module Index** – när du vill veta mer om en modul, t ex random eller copy
- **Tutorial** – förklaringar och exempel för grunderna i Python
- **Library reference** – kortfattad manual

Mer hjälp

- I Shell-fönstret finns snabb texthjälp.
Exempel:

```
>>dir(str)
>>help(str.center)
```
- Se även Referenser och FAQ på kursens webbsida!

Algoritm för sökning

1. Skapa en tom lista (*funna*) för alla träffar.
2. Gå igenom varje element i listan
 - Om elementet är det sökta:
Lägg till objektet till *funna*
3. Efteråt returnerar man *funna* med alla träffar.

Exempel:

```
ballader = sok("Bobby Ljunggren",listan)
```

Funktion för sökning

```
def sok(x, lista):
    """Söker efter låtskrivaren x i lista """
    funna = []
    for sang in lista:
        if (sang.text == x) or (sang.musik == x):
            funna.append(sang)
    return funna
```

Rekursion

- En rekursiv funktion i Python har:
 - Ett anrop av funktionen själv inuti funktionsdefinitionen.
 - Ett basfall som inte är rekursivt.
 - En if-sats som väljer mellan basfallet och det rekursiva anropet.
- Använd bara rekursion om du vet vad du gör!

Rekursivt exempel

```
def rek(x):
    if x<4:
        print x
        rek(x+1)
        print x, "igen!"

rek(1)
```

Granskning, exempel

- Funktion för centrerad utskrift:

```
... och vinnaren är ...
*****
Spaniens RODOLFO CHIKILICUATRE
      med låten
      BAILA EL CHIKI CHIKI
*****
```

Exempel (ej flexibelt)

```
def vinnarinfo(self):
    stjarnor = '*'*50
    print stjarnor
    print (self.land + "s " +
           self.artist.upper()).center(50)
    print "med låten".center(50)
    print (self.titel.upper()).center(50)
    print stjarnor
```

Flexibelt med parameter

```
def vinnarinfo(self, positioner):
    stjarnor = '*'*positioner
    print stjarnor
    print (self.land + "s " +
           self.artist.upper()).center(positioner)
    print "med låten".center(positioner)
    print
    print (self.titel.upper()).center(positioner)
    print stjarnor
```

Mer om granskning på grupptimmen!

Felhantering, t ex

- Felaktig inmatning:
 - Tecken istället för tal
 - För stort/för litet tal
- Filer:
 - Infil saknas
 - Felaktiga data i filen
- Lista/dictionary:
 - Index saknas
 - Nyckel saknas

Särfall - repetition

- När något blir fel i ett Python-program uppstår ett särfall, t ex `NameError`:

```
>>> print sko
```

Traceback (most recent call last):

```
File "<pyshell#17>", line 1, in -toplevel-  
    print sko
```

`NameError: name 'sko' is not defined`

- Man kan ta hand om särfall genom att införa `try-except-else`-satser för de delar i programmet som kan krascha.

särfall=exception

Exempel

```
def lasPengar():  
    """ Läser in tills man ger ett heltal """  
    pengar = None  
    while not pengar:  
        try:  
            svar = raw_input("Ange belopp: ")  
            pengar = int(svar)  
        except (ValueError), e:  
            print "Felaktigt belopp, försök igen"  
    return pengar
```

Grafiskt användargränssnitt (GUI)

Använd modulen **Tkinter**, som har klasser för komponenter.

Se "Referenser" på kursens webbsida: GUI-länkar

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

Hur gör man?

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 konstruktör 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.

Variabler

- Ett attribut som alla komponenter har är *variable*.
- Om man i förväg skapat ett variabelobjekt:
`s = StringVar()`
- så kan man koppla ihop variabel och komponent med
`knapp["variable"] = s`
- Metoden *get* hämtar data från en variabel.

Anropa funktion med knapptryck!

- Ett annat 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 metod med knapp så här:
`knapp["command"] = addera`

- Man kan också koppla ihop knappen med en funktion när den skapas:

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

Layout

- Komponenter har metoder som styr hur de ska placeras i fönstret.
- Enklast är att använda *pack*:
`knapp.pack()`
- Men bättre kontroll fås med *grid*:
`knapp.grid(row=4, column=3)`
- Rita först en skiss över hur det ska se ut!

	0	1	2	3
0	Rubrik			
1	Person:			
2	Sak:			
3	Verb:			
4	Adjektiv:	☐	☐	☐
5	Kroppsdel:	☐	☐	☐
6	Skriv saga			

.py eller .pyw?

- Program som är avsedda att startas med dubbelklick på ikonerna brukar man ge efternamnet ".pyw", t ex "saga.pyw"
- Då slipper man det svarta DOS-fönstret i bakgrunden.
- Men under avlusningen kan man utnyttja DOS-fönstret för kontrollutskrifter med `print` eller `raw_input`.

Tkinter och IDLE

- Din utvecklingsmiljö IDLE är skriven i Tkinter!
- Det innebär att *mainloop* redan är igång om du kör programmet inifrån IDLE, vilket kan få Shell-fönstret att bete sig underligt.
- Kommentera därför bort raden

```
roten.mainloop()
```


när du kör ditt Tkinter-program i IDLE.