

DD1311 Programmeringsteknik med PBL

Föreläsning 16

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.

Idag

- Några attribut hos GUI-komponenter:
 - `text`, `height`, `width`, `bg`, `font`
 - `variable`
 - `command`
- Lambda-uttryck
- Funktionsanrop med parameter

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 nån trycker på knapp så kan vi koppla ihop funktion 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()
```

Lambda-uttryck...

- Ett lambda-uttryck är ett alternativt sätt att skapa en funktion.
- Istället för att definiera funktionen på vanligt sätt


```
def dubbla(x):
    return 2*x
```
- Skriver man


```
dubbla = lambda(x) : 2*x
```

Parametrar då?

- I bägge varianterna ovan skickar man bara med *namnet* på funktionen.
- Hur ska man göra om funktionen har parametrar?
- Lösning: Använd *lambda-uttryck*!

...som parameter till Button

```
from Tkinter import *
def byttext(t):
    knapp["text"] = t
roten = Tk()
knapp = Button(roten,
               text="Tryck inte",
               command=(lambda: byttext("Aj!")))
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!
- Se programexemplet [saga.py](#)

	0	1	2	3
0	Rubrik			
1	Person:	<input type="text"/>		
2	Sak:	<input type="text"/>		
3	Verb:	<input type="text"/>		
4	Adjektiv:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	Kroppsdel:	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Skriv saga			

.py eller .pyw?

- Program som är avsedda att startas med dubbelklick på ikonen 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.