

# GUI 3

## Programmering 2

vt 25

## Widgets

Tidigare widgets

Widgeten Listbox

# Outline

## Widgets

Tidigare widgets

Widgeten Listbox

Knyta samman

# Widgets

## tidigare widgets

Vi har hittills gått igenom följande widgets:

- ▶ `Label(root, text='')` skapar en text-widget
- ▶ `Button(root, text='', command=funk)` skapar en knapp-widget. Tänk på att funktionen ska skrivas utan avslutande paranteser!
- ▶ `Entry(root)` ett fält användaren kan fylla i

`root` är i alla tre fallen variabeln som håller vårt fönster (`root = tk.Tk()`)

# Widgets

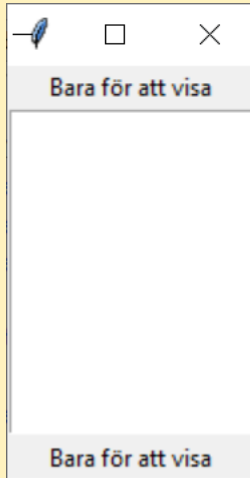
## Listbox

Till det här passet lägger vi till en ny widget: Listbox

```
1 import tkinter as tk
2
3 root = tk.Tk()
4 box = tk.Listbox(root)
5 text1 = tk.Label(root, text="Bara för att visa")
6 text2 = tk.Label(root, text="Bara för att visa")
7
8 text1.pack()
9 box.pack()
10 text2.pack()
11
12 root.mainloop()
```

# Widgets

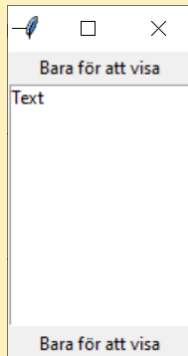
## Listbox



# Widgets

## Listbox

- ▶ För att lägga till i listan skriver du: `box.insert(tk.END, "Text")`
- ▶ `tk.END` anger att vi ska lägga till sist i listan



# Widgets

## Listbox

- ▶ För att läsa in vilka rader i listan som är markerade skriver du  
`selection = box.curselection()`
- ▶ Detta sparar en lista med alla index som är markerade i variabeln `selection`.
- ▶ Vill du bara ha den första som är markerad (eller enda) kan du skriva  
`selection[0]` eller `selection = box.curselection()[0]`



# Outline

## Widgets

- Tidigare widgets

- Widgeten Listbox

## Knyta samman

# Knyta samman

## Klasser

- ▶ När man vill göra ett större program är det ofta användbart att använda en klass för att skapa fönstret:
- ▶ En fördel med detta är att man kan komma åt alla instansvariabler i de olika metoderna fritt.

```
1 import tkinter as tk
2 class App():
3     def __init__(self):
4         self.root = tk.Tk()
5         self.root.mainloop()
6 app = App()
```

# Knyta samman

Ett program som läser in information från en fil

```
1 import tkinter as tk
2 class App():
3     def __init__(self):
4         self.root = tk.Tk() # Skapa fönstret
5         # Hantera widgets
6         self.create_widgets()
7         self.place_widgets()
8
9         self.root.mainloop() # Håll programmet vid liv
```

# Knyta samman

Ett program som läser in information från en fil

```
1  def create_widgets(self):  
2      """  
3      Metod som skapar alla widgets  
4      """  
5      self.file_entry = tk.Entry(self.root)  
6      self.load_file_button = tk.Button(self.root,  
7          text="Ladda fil", command=self.load_file)  
8      self.text_label = tk.Label(self.root, text="")
```

# Knyta samman

Ett program som läser in information från en fil

```
1  def place_widgets(self):
2      """
3      Metod som placerar ut alla widgets
4      """
5      self.file_entry.grid(row=0, column=0)
6      self.load_file_button.grid(row=0, column=1)
7      self.text_label.grid(row=1, column=0, columnspan=2)
```

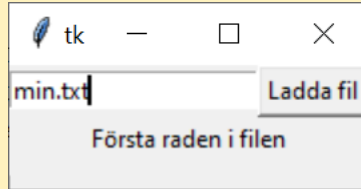
# Knyta samman

Ett program som läser in information från en fil

```
1  def load_file(self):
2      """
3      Metod som läser in en angiven fil och skriver ut första
4      raden
5      """
6      file_name = self.file_entry.get()
7      try:
8          with open(file_name, encoding="utf-8") as f:
9              self.data = f.readlines()
10     except FileNotFoundError: # Om filen inte finns
11         return # Avbryt det vi gör
12     self.text_label["text"] = self.data[0]
13
14 app = App() # Kör appen
```

# Knyta samman

Ett program som läser in information från en fil



# Knyta samman

Ett program som läser in information från en fil

## **App()**

```
file_entry: tk.Entry  
load_file_button: tk.Button  
text_label: tk.Label  
data: [str]
```

```
create_widgets(): void  
place_widgets(): void  
load_file(): void
```