

PROGRAMIRANJE U INTEGRISANIM TEHNOLOGIJAMA

Oznaka predmeta: PIT

Predavanje broj: 05

Nastavna jedinica: PYTHON,

Nastavne teme:

Mape, filteri, redukcija, lambda. Uvod u Tkinter. Dimenzije, boje, fontovi. Pakovanje widget-a, grid, place. Tkinter. Button. Checkbutton. Radiobutton. Label. Message. Canvas. Entry. Frame. Listbox. Menu. Scale. Scrollbar. LabelFrame. Spinbox. Text.

Predavač: prof. dr Perica S. Štrbac, dipl. ing.

Literatura:

Steven Lott: "Functional Python Programming", Packt Publishing, 2015.

Mape, filteri, redukcija, lambda

- Korišćenje mapiranja je oblika:

`map( funkcija, sekvenca )`

primeri:

posmatrajmo primer bez korišćenja map-a

```
>>> items = [1, 2, 3, 4, 5]
>>> squared = []
>>> for x in items:
    squared.append(x ** 2)
>>> squared
```

`[1, 4, 9, 16, 25]`

rešenje prethodnog primera uz upotrebu map-a

```
>>> items = [1, 2, 3, 4, 5]
>>> def sqr(x):
    return x ** 2
>>> list(map(sqr, items))
```

`[1, 4, 9, 16, 25]`

ili sa malo kompaktnijim kodom

```
>>> items = [1, 2, 3, 4, 5]
>>> list(map((lambda x: x**2), items))
```

`[1, 4, 9, 16, 25]`

Mape, filteri, redukcija, lambda

- Korišćenje liste funkcija kao sekvence (upotrebom lambde):

```
def square(x): return (x**2)
def cube(x):   return (x**3)
funcs = [square, cube]
for r in range(5):
    value = list(map(lambda x: x(r), funcs))
    print (value)
```

```
[0, 0]
[1, 1]
[4, 8]
[9, 27]
[16, 64]
```

- Korišćenje map-a je kao i petlje:

```
>>> def mymap(aFunc, aSeq):
    result = []
    for x in aSeq: result.append(aFunc(x))
    return result
```

```
>>> list(map(sqr, [1, 2, 3]))
```

```
#[1, 4, 9], sqr od ranije
```

```
>>> mymap(sqr, [1, 2, 3])
```

```
#[1, 4, 9]
```

Mape, filteri, redukcija, lambda

- Map je ugrađen tako da je brži od ručno kodovanog (my)map-a.
- Moguće je koristiti funkciju i liste za argumente mapa ako funkcija zahteva dva argumenta.

```
>>> pow(2,10)
1024
>>> pow(3,11)
177147
>>> pow(4,12)
16777216
>>>
>>> list(map(pow,[2, 3, 4], [10, 11, 12]))
[1024, 177147, 16777216]
```

- Pravljenje liste n-torki u Python-u 3x:

```
>>> m = [1,2,3]
>>> n = [1,4,9]
>>> def f(*x): return x;
>>> new_tuple = list(map(f, m, n))
>>> new_tuple
[(1, 1), (2, 4), (3, 9)]
```

Mape, filteri, redukcija, lambda

- Filteri
ekstrahuju svaki element sekvence za koga funkcija ima vrednost True.

```
>>> list(range(-5,5))  
[-5, -4, -3, -2, -1, 0, 1, 2, 3, 4]  
>>>  
>>> list( filter((lambda x: x < 0), range(-5,5)))  
[-5, -4, -3, -2, -1]
```

isto kao:

```
>>> result = []  
>>> for x in range(-5, 5):  
    if x < 0:  
        result.append(x)  
>>> result  
[-5, -4, -3, -2, -1]
```

- Redukcija
redukuje listu na jednu vrednost tako što primenjuje funkciju na elemente liste po paru argumenata i tako redom do kraja liste.

```
>>> from functools import reduce  
>>> reduce( (lambda x, y: x * y), [1, 2, 3, 4] )  
24
```

Mape, filteri, redukcija, lambda

- Ekvivalentan kod redukcije:

```
>>> L = [1, 2, 3, 4]
>>> result = L[0]
>>> for x in L[1:]:
>>>     result = result * x
>>> result
24
```

- Ekvivalentan kod redukcije kao funkcije:

```
>>> def myreduce(fnc, seq):
>>>     tally = seq[0]
>>>     for next in seq[1:]:
>>>         tally = fnc(tally, next)
>>>     return tally
>>> myreduce( (lambda x, y: x * y), [1, 2, 3, 4])
24
>>> myreduce( (lambda x, y: x / y), [1, 2, 3, 4])
0.041666666666666664
```

Mape, filteri, redukcija, lambda

- Npr. konkatencija string literala u rečenicu:

```
import functools
>>> L = ['Testing ', 'shows ', 'the ', 'presence', ' ', ' ', 'not ',
         'the ', 'absence ', 'of ', 'bugs']
>>> functools.reduce( (lambda x,y:x+y), L)
'Testing shows the presence, not the absence of bugs'
```

- Naravno, ovo može jednostavnije što smo koristili ranije:

```
>>> ''.join(L)
'Testing shows the presence, not the absence of bugs'
```

Ili korišćenjem operatora:

```
>>> import functools, operator
>>> functools.reduce(operator.add, L)
'Testing shows the presence, not the absence of bugs'
```

Mape, filteri, redukcija, lambda

- Lambda, primeri:

```
>>> def f (x): return x**2
```

```
...
```

```
>>> print (f(8))
```

```
64
```

```
>>>
```

```
>>> g = lambda x: x**2
```

```
>>>
```

```
>>> print (g(8))
```

```
64
```

```
>>> def make_incrementor (n): return lambda x: x + n
```

```
>>>
```

```
>>> f = make_incrementor(2)
```

```
>>> g = make_incrementor(6)
```

```
>>>
```

```
>>> print (f(42), g(42))
```

```
44 48
```

```
>>>
```

```
>>> print (make_incrementor(22)(33))
```

```
55
```

Mape, filteri, redukcija, lambda

```
>>> import functools
>>> foo = [2, 18, 9, 22, 17, 24, 8, 12, 27]
>>> print (list(filter(lambda x: x % 3 == 0, foo)))
                                                                [18, 9, 24, 12, 27]
>>> print (list(map(lambda x: x * 2 + 10, foo)))
                                                                [14, 46, 28, 54, 44, 58, 26, 34, 64]
>>> print (functools.reduce(lambda x, y: x + y, foo))
                                                                139

>>> nums = list(range(2, 50))
>>> for i in range(2, 8):
...     nums = list(filter(lambda x: x == i or x % i, nums))
...
>>> print (nums)
                                                                [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]
>>> sentence = 'It is raining cats and dogs'
>>> words = sentence.split(' ')
>>> print (words)
                                                                ['It', 'is', 'raining', 'cats', 'and', 'dogs']
>>> lengths = list(map(lambda word: len(word), words))
>>> print (lengths)
                                                                [2, 2, 7, 4, 3, 4]
```

Python, dimenzije, fontovi

- Dimenzije u oznakama jedinica: c (centimetri), i (inči), m (milimetri), p (tačka printera: 1/72").
- Boje: format "#fff" (bela) ili "#000000" (crna) ili "#000fff000" (zelena) ili po nazivima ("white", "black", "red", "green", "blue", "cyan", "yellow", "magenta").
- Fontovi: vrednost atributa font može biti i n-torka ("Times", "24", "bold italic")

```
import tkinter
```

```
from tkinter import font
```

```
top=tkinter.Tk(); font = font.Font(top, family="Verdana", size=50)
```

```
label=tkinter.Label(top, text="labela", font=font);
```

```
label.pack()
```

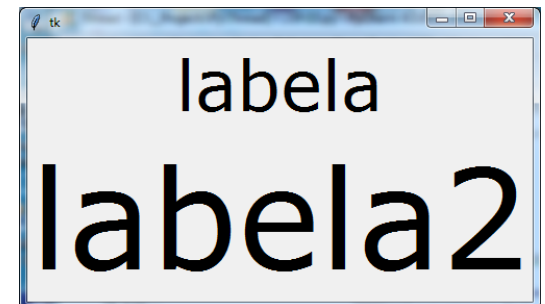
```
label2=tkinter.Label(top, text="labela2", font=("Verdana","100"));
```

```
label2.pack()
```

```
top.mainloop()
```

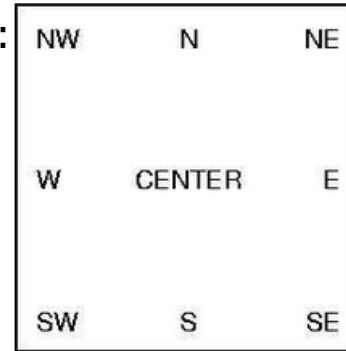
opcije su:

- | | |
|--------------|--|
| •family: | naziv fonta. |
| •size: | visina fonta u tačkama. |
| •weight: | debljina fonta "bold" ili "normal". |
| •slant: | ukošenost, "italic" ili "roman". |
| •underline: | 1 za podvlačen tekst, 0 za nepodvučen tekst. |
| •overstrike: | 1 za precrtan tekst, 0 za neprecrtan. |



Python, ankeri, reljefni stilovi

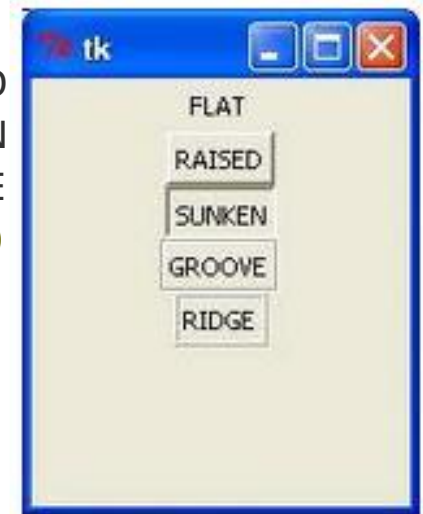
- Ankeri (anchors) imaju vrednost kao na slici:



- Reljefni stilovi (relief):

```
from tkinter import *      # tekuci prostor imena
import tkinter             # prostor imena tkinter
top = tkinter.Tk()

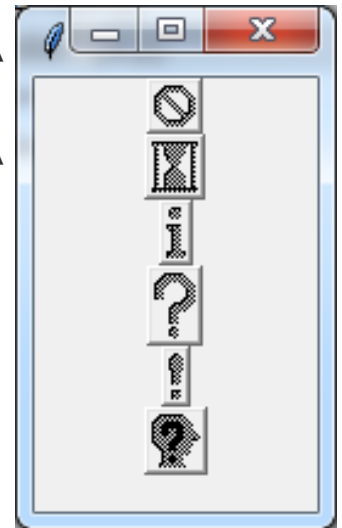
B1 = tkinter.Button(top, text = "FLAT", relief=FLAT )
B2 = tkinter.Button(top, text = "RAISED", relief=RAISED)
B3 = tkinter.Button(top, text = "SUNKEN", relief=SUNKEN)
B4 = tkinter.Button(top, text = "GROOVE", relief=GROOVE)
B5 = tkinter.Button(top, text = "RIDGE", relief=RIDGE )
B1.pack()
B2.pack()
B3.pack()
B4.pack()
B5.pack()
top.mainloop()
```



Python, bitmaps

- Bitmaps koje se pripremljene su: "error", "gray75", "gray50", "gray25", "gray12", "hourglass", "info", "questhead", "question", "warning".

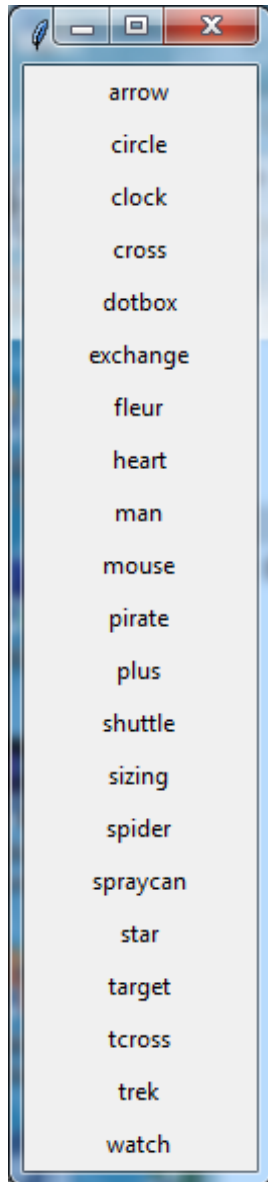
```
from tkinter import *
import tkinter
top = tkinter.Tk()
B1 = tkinter.Button(top, text ="error", relief=RAISED, bitmap="error")
B2 = tkinter.Button(top, text ="hourglass", relief=RAISED,\
bitmap="hourglass")
B3 = tkinter.Button(top, text ="info", relief=RAISED, bitmap="info")
B4 = tkinter.Button(top, text ="question", relief=RAISED,\
bitmap="question")
B5 = tkinter.Button(top, text ="warning", relief=RAISED,\
bitmap="warning")
B6 = tkinter.Button(top, text ="warning", relief=RAISED,\
bitmap="questhead")
B1.pack()
B2.pack()
B3.pack()
B4.pack()
B5.pack(); B6.pack();
top.mainloop()
```



Python, kursori

Kursori:

"arrow"
"circle","clock"
"cross"
"dotbox"
"exchange"
"fleur"
"heart","man"
"mouse"
"pirate"
"plus"
"shuttle"
"sizing"
"spider"
"spraycan"
"star"
"target"
"tcross"
"trek"
"watch"



```
from tkinter import *  
import tkinter  
top = tkinter.Tk()  
Buttons=[]  
Cursors=["arrow", "circle", "clock", "cross",  
"dotbox", "exchange", "fleur", "heart", "man",  
"mouse", "pirate", "plus", "shuttle", "sizing",  
"spider", "spraycan", "star", "target", "tcross",  
"trek", "watch"]  
for c in range (len(Cursors)):  
    Buttons.append(tkinter.Button(top, \  
        text=Cursors[c], \  
        relief=FLAT, cursor=Cursors[c])).pack()  
top.mainloop()
```

Python, pakovanje widget-a

- Pakovanje komponenti ide tako da geometry manager organizuje widget-e u blokovima pre njihovog postavljanja u roditeljski widget.
- Koristi se: `widget.pack( pack_options )` gde su opcije kao što sledi:
 - `expand`: ako je `true` widget se širi na ceo slobodan prostor roditeljskog widget-a.
 - `fill`: popunjavanje prostora widget-a (`NONE` (podrazumevano), `X` (horizontalno popunjavanje), `Y` (vertikalno popunjavanje), `BOTH` (i horizontalno i vertikalno popunjavanje).
 - `side`: strana roditeljskog prozora gde se pakuje widget: `TOP` (podrazumevano), `BOTTOM`, `LEFT` ili `RIGHT`.

```
from tkinter import *
root = Tk()
frame = Frame(root)
frame.pack()
bottomframe = Frame(root)
bottomframe.pack( side = BOTTOM )
redbutton = Button(frame, text="Red", fg="red")
redbutton.pack( side = LEFT)
greenbutton = Button(frame, text="Brown", fg="brown")
greenbutton.pack( side = LEFT )
```

Python, pakovanje widget-a, grid

```
bluebutton = Button(frame, text="Blue", fg="blue")
bluebutton.pack( side = LEFT )
blackbutton = Button(bottomframe, text="Black", fg="black")
blackbutton.pack( side = BOTTOM)
root.mainloop()
```



- Grid omogućuje matrično raspoređivanje

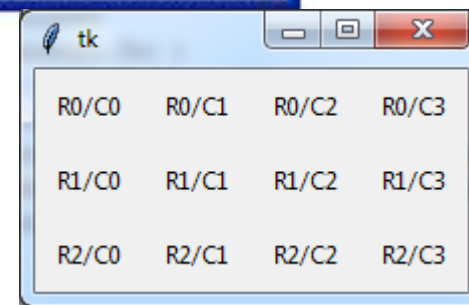
`widget.grid( grid_options )`

gde su opcije:

- column : kolona u kom je widget (podrazumevano 0, krajnja leva kolona).
- columnspan: na koliko se kolona proteže widget (podrazumevano 1).
- padx, pady : unutrašnje margine widget-a.
- padx, pady : vanjske margine widget-a.
- row: red u kom je widget.
- rowspan : protezanje widget-a po redovima (podrazumevano 1).
- sticky : pozicija widget-a u ćeliji (podrazumevano centrirano, vrednost '').
ostale pozicije su prema kompasu: N, E, S, W, NE, NW, SE i SW.

Python, grid, place

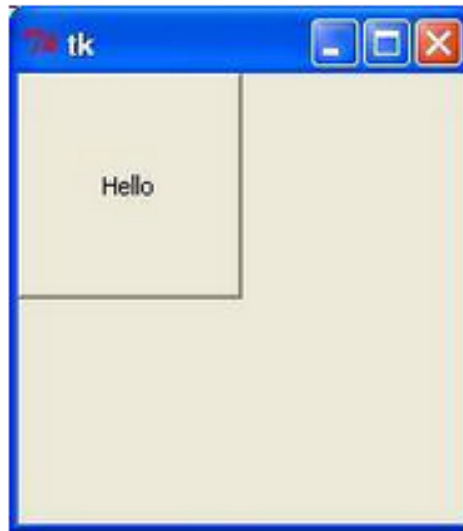
```
import tkinter
root = tkinter.Tk( )
for r in range(3):
    for c in range(4):
        tkinter.Label(root, text='R%s/C%s'%(r,c),
            borderwidth=1 ).grid(row=r,column=c)
root.mainloop( )
```



- Postavljanje widget-a na specifičnu poziciju u roditeljskom widget-u:
 `widget.place( place_options )`
 opcije su:
 - `anchor` : N, E, S, W, NE, NW (podrazumevano), SE ili SW.
 - `bordermode` : INSIDE (podrazumevano), OUTSIDE.
 - `height, width` : visina i širina u pikselima.
 - `relheight, relwidth` : normalizovano prema veličini roditeljskog widgeta [0,1].
 - `relx, rely` : normalizovani horizontalni i vertikalni ofset prema veličini roditeljskog widget-a.
 - `x, y` : horizontalni i vertikalni ofset u pikselima.

Python, place

```
import tkinter
from tkinter import messagebox
from tkinter import *
top = tkinter.Tk()
def helloCallBack():
    messagebox.showinfo( "Hello Python", "Hello World")
B = tkinter.Button(top, text ="Hello", command = helloCallBack)
B.pack()
B.place(height=100, width=100)
top.mainloop()
```



Python, GUI

- Python obezbeđuje razvoj grafičkog programskog interfejsa (GUIs).
- Slede neki od GUI interfejsa:
 - Tkinter: je Python interfejs za Tk GUI alat koji se isporučuje uz Python.
 - wxPython: je open-source Python interfejs za wxWindows
<http://wxpython.org>.
 - JPython: JPython je Python port za Javu gde se korišćenjem Python skriptova pristupa Java (class) bibliotekama na lokalnom računaru <http://www.jython.org>.

Tkinter programiranje

- Tkinter je standardna GUI biblioteka za Python.
- Tkinter obezbeđuje moćan objektno orijentisan interfejs.
- Da bi se kreirala GUI aplikacija korišćenjem Tkinter-a potrebno je da se uradi kao što sledi:
 - Import modula tkinter.
 - Kreiranje glavnog prozora GUI aplikacije.
 - Dodavanje widget-a u GUI aplikaciju.
 - Ulazak u petlju za osluškivanje događaja koje pokreće korisnik.

Python, tkinter

- Primer:

```
#!/usr/bin/python
import tkinter
top = tkinter.Tk()
# Code to add widgets will go here...
top.mainloop()
```



Tkinter widget-i

- Tkinter obezbeđuje različite kontrole (buttons, labels, text boxes ...) koje se koriste u GUI aplikacijama. Ove kontrole se nazivaju widget-ima.
- Slede opisi widget-a:
 - Button koristi se za prikaz programskog dugmeta.
 - Canvas omogućuje crtanje oblika (lines, ovals, polygons, rectangles).
 - Checkbutton prikazuje više opcija od kojih se može odabrati bilo koja kombinacija.
 - Entry koristi se za prikaz polja za unos linije teksta i prihvatanje vrednosti koju unese korisnik.
 - Frame koristi se kao kontejner za organizaciju drugih widget-a.
 - Label koristi se da obezbedi liniju naslova widget-a. Može sadržavati sliku.

Python, tkinter

- Listbox koristi se da omogući listu opcija za korisnika.
- Menubutton dugme za meni
- Menu meni.
- Message koristi se za prikaz višelinijskog tekst polja.
- Radiobutton prikazuje opcije kao radio dugmad pri čemu korisnik može da selektuje samo jednu opciju.
- Scale obezbeđuje korišćenje slajdera.
- Scrollbar skrolovanje različitih widget-a kao npr. list box-ova.
- Text prikazuje višelinijski tekst.
- Toplevel obezbeđuje odvojeni window kontejner.
- Spinbox varijanta entry widget-a koji se koristi da selektuje jednu od fiksnog broja vrednosti.
- PanedWindow predstavlja kontejner widget koji može da sadrži bilo koji broj pan-ova aranžiranih horizontalno ili vertikalno.
- LabelFrame je jednostavni kontejner. Koristi se za razmak ili kontejner za složene rasporede.
- messagebox prikazuje poruku u aplikaciji.

Python, Button

```
w = Button ( master, option=value, ... )
```

parameteri su kao što sledi:

- master: roditeljski prozor.
- options: opcije, npr:
 - activebackground boja pozadine dugmeta pod kursorom.
 - activeforeground boja teksta dugmeta pod kursorom.
 - bd border u pikselima (podrazumevano 2).
 - bg normalna boja pozadine.
 - **command** klik funkcija.
 - fg normalna boja teksta.
 - font font labele dugmeta.
 - height visina dugmeta u linijama teksta za tekstualno dugme ili pikselima za dugme sa slikom.
 - image slika za dugme.

Python, Button

- `justify` poravnanje više linija teksta (LEFT, CENTER, RIGHT).
- `padx` unutrašnje margine za tekst (left,right).
- `pady` unutrašnje margine za tekst (up,down).
- `relief` tip bordera (SUNKEN, RAISED, GROOVE, RIDGE...).
- `state` podrazumevno stanje je NORMAL, može biti pasivno (DISABLED), aktivno (ACTIVE, kada je miš iznad dugmeta).
- `underline` podrazumevano je -1, a ako je nenegativno onda pokazuje koji je indeks karaktera u tekstu koji je podvučen.
- `width` širina dugmeta u slovima (ako prikazuje tekst) odnosno pikselima (ako prikazuje sliku)
- `wraplength` širina za namotavanje teksta (wrap) tako da će tekst biti namotan u okviru date širine.

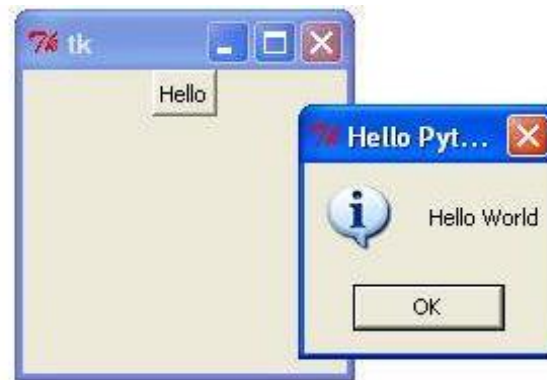
Python, Button

- Metode
 - `flash()` dugme nekoliko puta blinka menjajući aktivnu i normalnu boju. Ostavlja dugme u originalnom stanju. Ignoriše se ako je dugme onemogućeno.
 - `invoke()` poziva callback funkciju dugmeta i vraća što vraća ta funkcija. Nema efekta ako je dugme onemogućeno ili nema callback funkcije.

```
import tkinter
from tkinter import messagebox
```

```
top = tkinter.Tk()
def helloCallBack():
    messagebox.showinfo( "Hello Python", "Hello World")
```

```
B = tkinter.Button(top, text ="Hello", command = helloCallBack)
B.pack()
top.mainloop()
```



Python, Checkbutton

```
w = Checkbutton ( master, option, ... )
```

parameteri:

- master: roditeljski prozor.
- opcije (navode se kao CSV parovi key=value), slede primeri:
 - activebackground boja pozadine kada je checkbutton ispod kursora.
 - activeforeground boja pisanja kada je checkbutton ispod kursora.
 - bg normalna boja pozadine koja se prikazuje iza lbele i indikatora.
 - bitmap prikaz slike na dugmetu.
 - bd veličina bordera oko indikatora (podrazumevano 2px).
 - **command** procedura koja se poziva kada korisnik promeni stanje checkbutton-a.
 - cursor promena izgleda kursora (arrow, dot, ...) kada je kursor iznad checkbutton-a.
 - disabledforeground boja onemogućenog checkbutton-a.
 - font font teksta.
 - fg boja teksta.

Python, Checkbutton

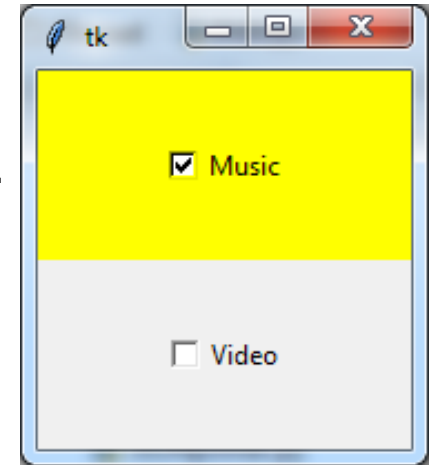
- height broj linija teksta na checkbutton-u (podrazumevano 1).
- image prikaz slike.
- justify poravnanje teksta (CENTER, LEFT, RIGHT).
- offvalue postavljanje vrednosti za isključen checkbutton (podrazumevano je 0).
- onvalue postavljanje vrednosti za uključen checkbutton (podrazumevano je 1).
- padx unutrašnja leva i desna margina (podrazumevano 1px).
- pady unutrašnja gornja i donja margina (podrazumevano 1px).
- relief podrazumevano relief=FLAT
- selectcolor boja setovanog checkbutton-a (podrazumevano "red").
- selectimage slika se pojavljuje ako je checkbutton setovan.

Python, Checkbutton

- **state** podrazumevano state=NORMAL, može se koristiti state=DISABLED za onemogućavanje (posiveće). Ako je kursor iznad checkbutton-a state=ACTIVE.
- **text** tekst labele checkbutton-a (za tekst u više redova koristiti "\n").
- **underline** podvlačenje karaktera (podrazumevano -1, nema podvlačenja) teksta labele. Indeks broj od nule označava koji će karakter teksta biti podvučen.
- **variable** kontrolna varijabla koja se odnosi na tekuće stanje checkbutton-a. (varijabla je **IntVar** sa vrednostima 0 (nesetovan) i 1 (setovan).
- **width** podrazumevana širina checkbutton-a je određena veličinom slike ili teksta. Širina se može postaviti kao broj karaktera.
- **wraplength** podrazumevano linije teksta se ne namotavaju. Može se postaviti vrednost koja označava broj karaktera koji se odnosi na prelom linije.

Python, Checkbutton

- Metode:
 - select() uključuje checkbutton.
 - deselect() isključuje checkbutton.
 - toggle() menja stanje uključenosti checkbutton — a.
 - flash() checkbutton blinka aktivnom i normalnom bojom te ostaje početna boja.
 - invoke() kao da je korisnik kliknuo da promeni stanje.



```
from tkinter import *
import tkinter
def pritisnuo():
    print(CheckVar1.get())
top = tkinter.Tk()
CheckVar1 = IntVar(); CheckVar2 = IntVar()
C1 = Checkbutton(top, bg="yellow", text = "Music", variable = CheckVar1, \
                  onvalue = 1, offvalue = 0, height=5, \
                  width = 20, command=pritisnuo)
C2 = Checkbutton(top, text = "Video", variable = CheckVar2, \
                  onvalue = 1, offvalue = 0, height=5, \
                  width = 20)
C1.pack();C2.pack(); top.mainloop()
```

Python, Radiobutton

```
w = Radiobutton ( master, option, ... )
```

Parametri:

- master: roditeljski prozor.
- options: date u CSV formi parova key=value.

Primeri opcija su kao što sledi:

- activebackground boja pozadine kada je miš iznad radiobutton-a.
- activeforeground boja pisanja kada je miš iznad radiobutton-a.
- anchor poravnanje teksta ako widget ima više prostora od prostora teksta (podrazumevano anchor=CENTER).
- bg boja pozadine.
- bitmap prikaz slike.
- borderwidth širina border-a oko indikatora (podrzumevano 2px).
- **command** procedura koja se poziva pri promeni stanja radiobutton-a.
- cursor tip kursora iznad radiobutton-a.
- font font teksta.

Python, Radiobutton

- fg boja pisanja.
- height visina radiobutton-a u linijama teksta (podrazum. 1).
- image prikaz slike.
- justify poravnanje teksta (CENTER, LEFT, RIGHT).
- padx unutrašnja leva i desna margina (podrazumevano 1).
- pady unutrašnja gornja i donja margina (podrazumevano 1).
- relief dekorativni border oko labele (podrazumevano FLAT).
- selectcolor boja selektovanog radiobutton-a (podrazumevano red).
- selectimage slika na selektovanom radiobutton-u.
- state podrazumevano state=NORMAL (state=DISABLED, state=ACTIVE).
- text tekst labele pored radiobutton-a (uračunava "\n").
- **textvariable** veza teksta sa datom varijablom tipa **StringVar**.
- underline podvlačenje n-tog slova (podrazumevano underline=-1).

Python, Radiobutton

- `value` vrednost radiobutton-a u grupi (ako je kontrolna varijabla `IntVar`, dodeljuje se broj, a ako je kontrolna varijabla `StringVar` dodeljuje se string).
- **`variable`** varijabla koju dele radiobutton-i u grupi (**`IntVar`** ili **`StringVar`**).
- `width` širina labele u karakterima (ako nije data veličina odgovara sadržaju).
- `wraplength` prelom linije na datoj dužini u karakterima (podrazumevano 0, prelom linije je na `"\n"`).

Metode:

- `select()` uključuje (turns on) the radiobutton.
- `deselect()` isključuje (turns off) radiobutton.
- `flash()` radiobutton zablinka (aktivnom i normalnom bojom).
- `invoke()` programski klik na radiobutton.

Python, Radiobutton

```
from tkinter import *
def sel():
    selection = "You selected the option " + str(var.get())
    label.config(text = selection)
root = Tk()
var = IntVar()
R1 = Radiobutton(root, text="Option 1", variable=var, value=1, command=sel)
R1.pack( anchor = W )
R2 = Radiobutton(root, text="Option 2", variable=var, value=2, command=sel)
R2.pack( anchor = W )
R3 = Radiobutton(root, text="Option 3", variable=var, value=3, command=sel)
R3.pack( anchor = W )
label = Label(root)
label.pack()
root.mainloop()
```



Python, Canvas

```
w = Canvas ( master, option=value, ... )
```

parameteri su kao što sledi:

- master: roditeljski prozor.
- opcije (mogu se navoditi kao CSV parovi key,value), npr:
 - bd border u pikselima (podrazumevano 2).
 - bg normalna boja pozadine.
 - cursor kursor koji se koristi u kanvasu (arrow, circle, dot...).
 - height y dimenzija veličine kanvasa.
 - relief tip bordera (SUNKEN, RAISED, GROOVE, RIDGE).
 - scrollregion n — torka (w, n, e, s) koja definiše kolika se površina može skrolovati.
 - width x dimenzija veličine kanvasa.

Python, Canvas

- `xscrollincrement` vrednost skrolovanja po x osi.
 - `xscrollcommand` linkovanje ka horizontalnom scroll bar-u.
 - `yscrollincrement` vrednost skrolovanja po y osi.
 - `yscrollcommand` linkovanje ka vertikalnom scroll bar-u.
- Neki elementi kanvasa:
 - `create_arc` kreira luk.
`coord = 10, 50, 240, 210`
`arc = canvas.create_arc(coord,start=0,extent=150,fill="blue")`
 - `create_image` prikazuje sliku (instanca klase: `BitmapImage` ili `PhotoImage`).
`filename = PhotoImage(file = "sunshine.gif")`
`image = canvas.create_image(50,50, anchor=NE, image=filename)`
 - `create_line` crta liniju.
`line = canvas.create_line(x0,y0, x1,y1, ..., xn,yn, options)`

Python, Canvas

- `create_oval` crta krug ili elipsu na datim koordinatama (dva para koordinata gornji levi ugao i donji desni ugao okružujućeg pravougaonika).

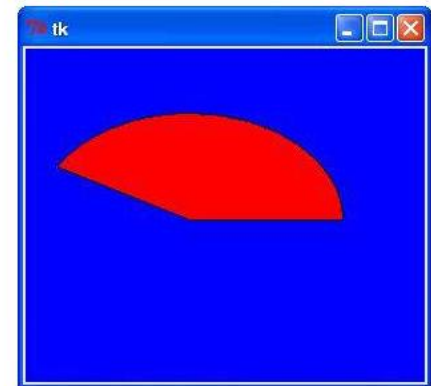
```
oval = canvas.create_oval(x0, y0, x1, y1, options)
```

- `create_polygon` crta poligon sa najmanje tri vrha.

```
oval = canvas.create_polygon(x0,y0, x1,y1,...xn, yn, options)
```

Primer:

```
import tkinter
top = tkinter.Tk()
C = tkinter.Canvas(top, bg="blue", height=250, width=300)
coord = 10, 50, 240, 210
arc = C.create_arc(coord, start=0, extent=150, fill="red")
C.pack()
top.mainloop()
```



Python, Entry

- Entry widget se koristi za prihvatanje jedne linije teksta korisnika.
 - Ako se želi prikaz više linija teksta koji se može editovati onda bi trebalo koristiti widget Text .
 - Ako se želi prikaz jedne ili više linija teksta koje korisnik ne može modifikovati onda bi trebalo koristiti widget Label.

```
w = Entry( master, option, ... )
```

Parametri:

- master: roditeljski prozor.
- options: u CSV formi parova key=value.
- Slede primeri opcija:
 - bg normalna boja pozadine (iza labele i indikatora).
 - bd border oko indikatora (podrazumevano 2px).
 - cursor oblik kursora (arrow, dot, ...) kada je miš iznad widget-a.
 - font font teksta.

Python, Entry

- exportselection eksport selektovanog teksta u clipboard (podrazum. 1).
- fg boja pisanja.
- justify poravnanje teksta (CENTER, LEFT, RIGHT).
- relief podrazumevano FLAT, widget nije van svoje pozadine.
- selectbackground boja pozadine selektovanog teksta.
- selectborderwidth border selektovanog teksta (podrazumevano 1px).
- selectforeground boja pisanja selektovanog teksta.
- show određuje prikaz karaktera (npr. za lozinku show="*").
- state stanje (podrazumevano state=NORMAL, može biti state=DISABLED i state=ACTIVE).
- **textvariable** veza teksta sa datom varijablom tipa **StringVar**.
- width širina widget-a daje se u karakterima (podrazumevano odgovara širini teksta).
- xscrollcommand linkovanje ka scrollbar-u ako se očekuje duži tekst od prostora za ispis.

Python, Entry

Metode:

- `delete ( first, last=None )` brisanje karaktera od first indeksa do isključno last indeksa. Ako je drugi parametar izostavljen briše se samo karakter čiji je indeks određen parametrom first.
- `get()` vraća uneseni tekst kao string.
- `icursor ( index )` postavljanje kursora za umetanje pre karaktera čiji je indeks index.
- `index ( index )` pomeranje sadržaja tako da karakter datog indeksa bude krajnji levi vidljivi karakter. Nema efekta ako je sav tekst u entry widget-u.
- `insert ( index, s )` ubacuje string s pre karaktera datog indeksa.
- `select_adjust ( index )` osiguranje da selekcija uključuje karakter datog indeksa.
- `select_clear()` briše selekciju. Ako nema selekcije onda nema efekta.
- `select_from ( index )` postavljanje ANCHOR indeksa i selekcija pripadnog karaktera.
- `select_present()` vraća True ako postoji selekcija, inače vraća False.

Python, Entry

- `select_range ( start, end )` selekcija teksta od startnog indeksa (start) do isključno završnog indeksa (end). Startna pozicija mora biti pre krajnje pozicije.
- `select_to ( index )` selekcija karaktera od ANCHOR pozicije do isključno index pozicije.
- `xview ( index )` skrolovanje da indeks *index* bude vidljiv.
- `xview_scroll ( number, what )` horizontalno skrolovanje. Argument *what* može biti UNITS (skrolovanje za širinu datu u karakterima) ili PAGES (skrolovanje za veličinu entry widget-a). Ako je parametar *number* pozitivan skroluje se desno a ako je negativan onda levo.

```
from tkinter import *
```

```
import tkinter
```

```
def prikazi():
```

```
    print(em.get())
```

```
top = tkinter.Tk()
```

```
L1 = Label(top, text="User Name"); L1.pack( side = TOP); em = StringVar()
```

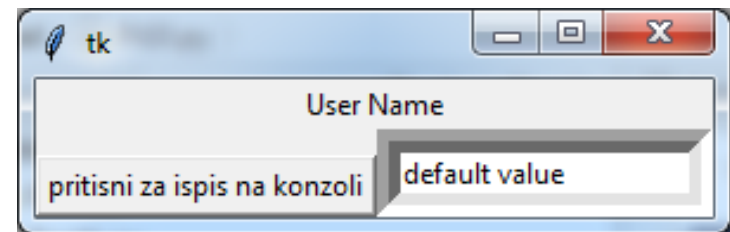
```
E1 = Entry(top, bd =10, textvariable=em); em.set("default value")
```

```
E1.pack(side = RIGHT)
```

```
B1 = Button(top, text="pritisni za ispis na konzoli", command=prikazi)
```

```
B1.pack(side = BOTTOM)
```

```
top.mainloop()
```



Python, Frame

```
w = Frame ( master, option, ... )
```

- Parametri:
 - master: roditeljski prozor.
 - options: u CSV formi parova key=value.
- Primeri opcija su:
 - bg normalna boja pozadine iza labele i indikatora.
 - bd veličina border-a oko indikatora (podrazumevano 2px).
 - cursor tip kursora (arrow, dot, ...) kada je miš iznad komponente.
 - height visina frame-a.
 - relief podrazumevano relief=FLAT, komponenta nije van njene pozadine.
 - width podrazumevana širina odgovara sadržaju, a može se postaviti na datu vrednost.

Python, Frame

```
from tkinter import *  
root = Tk()
```

```
frame = Frame(root)  
frame.pack()
```

```
bottomframe = Frame(root)  
bottomframe.pack( side = BOTTOM )
```

```
redbutton = Button(frame, text="Red", fg="red")  
redbutton.pack( side = LEFT)  
greenbutton = Button(frame, text="Brown", fg="brown")  
greenbutton.pack( side = LEFT )  
bluebutton = Button(frame, text="Blue", fg="blue")  
bluebutton.pack( side = LEFT )
```

```
blackbutton = Button(bottomframe, text="Black", fg="black")  
blackbutton.pack( side = BOTTOM)
```

```
root.mainloop()
```



Python, Label

```
w = Label ( master, option, ... )
```

Parametri:

- master: roditeljski prozor.
- options: opcije u CSV formi parova key=value.
- Slede primeri opcija:
 - anchor pozicija teksta ako widget ima više prostora nego što ga tekst zauzima (podrazumevano CENTER).
 - bg normalna boja pozadine.
 - bitmap za labelu koja prikazuje sliku.
 - bd granica oko indikatora (podrazumevano 2px).
 - cursor oblik kursora kada je iznad.
 - font font teksta labele.
 - fg boja teksta labele.
 - height vertikalna dimenzija.
 - image prikaz statičke slike (image object).
 - justify poravnanje teksta (LEFT, CENTER (podrazumevano), RIGHT).

Python, Label

- padx unutrašnja leva i desna margina (podrazumevano 1).
- pady unutrašnja gornja i donja marginav (podrazumevano 1).
- relief dekorativni border (podrazumevano je FLAT).
- text prikaz jedne ili više linija teksta u labeli (korišćenjem "\n" forsira se nova linija).
- **textvariable** postavljanje teksta kao kontrolne varijable klase **StringVar**.
- underline postavljanje podvlačenja (_) ispod n-tog slova teksta indeksiranog od nule (podrazumevano -1).
- width širina labele u karakterima (za fit-ovanje konteksta).
- wraplength ograničavanje broja karaktera po liniji (podrazumevano 0, nema preloma linije teksta osim ako je "\n").

```
from tkinter import *  
root = Tk()  
var = StringVar()  
label = Label( root, textvariable=var, relief=RAISED )  
var.set("Hey!? How are you doing?")  
label.pack()  
root.mainloop()
```



Python, Listbox

```
w = Listbox ( master, option, ... )
```

Parametri:

- master: roditeljski prozor.
- opcije: formatirane kao SCV parovi key=value.
- Opcije su kao što sledi:
 - bg normalna boja pozadine.
 - bd veličina border-a (podrazumevano 2px).
 - cursor tip kursora iznad listbox-a.
 - font font listbox-a.
 - fg boja teksta u listbox-u.
 - height broj stavki u listbox-u (podrazumevano 10).
 - relief 3D border senčenje (podrazumevano SUNKEN).
 - selectbackground boja pozadine selektovanog teksta.
 - selectmode određuje koliko je stavki moguće selektovati i kako ih selektuje miš kao što sledi:

Python, Listbox

- BROWSE: selektovanje jedne stavke listbox-a (podrazumevano) ali se selektovanje može raditi prevlačenjem mišem.
- SINGLE: selektovanje jedne stavke listbox-a i ne može se vršiti prevlačenjem mišem.
- MULTIPLE: selektovanje više linija.
- EXTENDED: selekcija grupe susednih stavki klikom na prvu i prevlačenjem do poslednje stavke.
- width širina u karakterima (podrazumevano 20).
- xscrollcommand za horizontalno skrolovanje listbox treba linkovati sa horizontalnim skrolbarom.
- yscrollcommand za vertikalno skrolovanje listbox treba linkovati sa horizontalnim skrolbarom.

Metode Listbox objekta:

- activate (index) selekcija indeksom specificirane stavke.
- curselection() vraća n-torku indeksa selektovanih stavki (broji od nultog indeksa).
Ako nema selektovanih stavki vraća praznu n-torku.

Python, Listbox

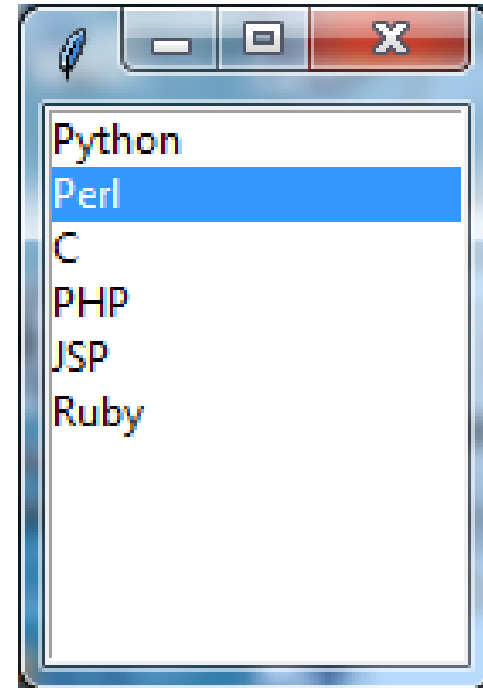
- `delete ( first, last=None )` briše stavke u navedenom opsegu indeksa (ako je drugi argument izostavljen briše se samo stavka specificirana prvim indeksom).
- `get ( first, last=None )` vraća n-torku koja sadrži tekstove stavki navedenog indeksnog opsega.
- `index ( i )` postavljanje stavke indeksa i na vrh widget-a.
- `insert ( index, *elements )` ubacivanje jedne ili više stavki u listbox pre specificirane stavke čiji je indeks dat prvim parametrom. Za dodavanje na kraj za prvi argument se uzima vrednost END.
- `nearest ( y )` vraća indeks vidljive stavke najbliže y-koordinati y.
- `see ( index )` podešavanje da stavka datog indeksa bude vidljiva.
- `size()` vraća broj linija u listbox-u.
- `xview()` za postizanje horizontalne skrolabilnosti postavlja se command opcija asociranog horizontalnog scrollbar-a na ovaj metod.

Python, Listbox

- `xview_scroll ( number, what )` horizontalno skrolovanje listbox-a. Argument `number` odnosi se na koliko se skroluje, a argument `what` određuje da li se skroluje u karakterima (UNITS) ili u stranicama (PAGES) tj. za širinu listbox-a.
- `yview()` za pravljenje vertikalne skrolabilnosti postavlja se komandna opcija asociranog vertikalnog scrollbar-a na ovaj metod.
- `yview_scroll ( number, what )` vertikalno skrolovanje listbox-a. Argument `number` odnosi se na koliko se skroluje, a argument `what` određuje da li se skroluje u karakterima (UNITS) ili u stranicama (PAGES) tj. za visinu listbox-a.

Python, Listbox

```
from tkinter import *
import tkinter
top = Tk()
Lb1 = Listbox(top, selectmode=SINGLE )
Lb1.insert(0, "Python")
Lb1.insert(1, "Perl")
Lb1.insert(2, "C")
Lb1.insert(3, "PHP")
Lb1.insert(4, "JSP")
Lb1.insert(5, "Ruby")
Lb1.activate(1)
Lb1.select_set(1,1)
Lb1.pack()
#print(str(Lb1.get(ACTIVE)))
print(str(Lb1.get(12)))
top.mainloop()
```



Python, Message

```
w = Message ( master, option, ... )
```

Parametri:

- master: roditeljski prozor.
- options: date u CSV formi parova key=value.
- Opcije su kao što sledi:
- anchor poravnanje teksta ako widget ima više prostora od prostora teksta (podrazumevano anchor=CENTER).
- bg boja pozadine.
- bitmap prikaz slike.
- bd veličina bordera oko indikatora (podrazumevano 2px).
- cursor tip kursora kada je miš iznad komponente.
- font font teksta.
- fg boja teksta.
- height visina.
- image prikaz statične slike.
- justify poravnanje teksta (LEFT, CENTER, RIGHT).

Python, Message

- `padx` unutrašnja leva i desna margina (podrazumevano 1).
- `pady` unutrašnja gornja i donja margina (podrazumevano 1).
- `relief` dekorativni border oko labele (podrazumevano FLAT).
- `text` tekst labele (string kod koga se uračunavaju znaci "\n").
- `textvariable` tekst labele je u kontrolnoj varijabli klase `StringVar`.
- `underline` podvlačenje slova teksta indeksirano od 0 (podrazumevano -1).
- `width` širina labele u karakterima (ako nije postavljena veličina odgovara sadržaju).
- `wraplength` prelom teksta na datom broju karaktera (podrazumevano 0).
automatski wrap-uje tekst

```
from tkinter import *
root = Tk()
var = StringVar()
label = Message( root, textvariable=var, relief=RAISED )
var.set("Hey!? How are you doing?")
label.pack()
root.mainloop()
```



Python, messagebox

```
messagebox.FunctionName(title, message [, options])
```

Parametri:

- FunctionName: naziv pripadne funkcije koja se veže za message box.
- title: naslov messagebox-a.
- message: tekstualna poruka.
- options: opcije

Metode:

- showinfo()
- showwarning()
- showerror()
- askquestion() (vraća yes no)
- askyesno () (vraća True False)
- askokcancel() (vraća True False)
- askretrycancel () (vraća True False)

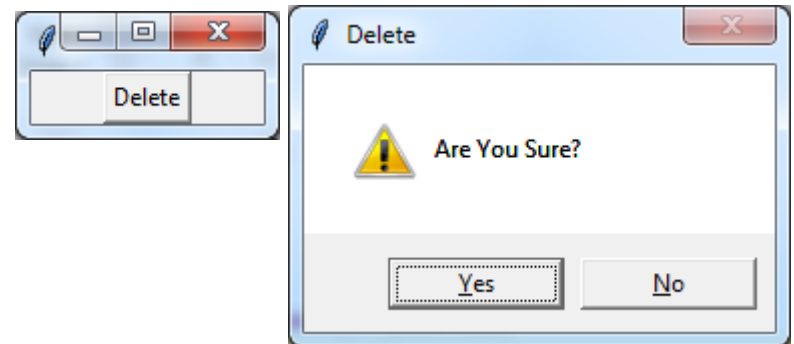
Python, messagebox

```
import tkinter
from tkinter import messagebox
top = tkinter.Tk()
def hello():
    messagebox.showinfo("Say Hello", "Hello World")
B1 = tkinter.Button(top, text = "Say Hello", command = hello)
B1.pack()
top.mainloop()
```



```
import tkinter
from tkinter import messagebox
top = tkinter.Tk()
def deleteme():
    answer = messagebox.askquestion("Delete", "Are You Sure?",
                                    icon='warning')

    if answer=='yes':
        print ("Deleted")
    else:
        print ("I'm Not Deleted Yet")
B1 = tkinter.Button(top, text = "Delete",
                    command = deleteme)
B1.pack()
top.mainloop()
```



Python, Menu

```
w = Menu ( master, option, ... )
```

Parametri:

- master: roditeljski prozor.
- options: opcije u CSV formi parova key=value.
- Opcije su kao što sledi:
 - activebackground boja pozadine kada je opcija ispod miša.
 - activeborderwidth širina bordera oko opcije koja je ispod miša (podrazumevano 1px).
 - activeforeground boja pisanja opcije koja je ispod miša.
 - bg boja pozadine.
 - bd širina bordera oko pocija (podrazumevano 1).
 - cursor tip kursora iznad opcije.
 - disabledforeground boja pisanja onemogućene stavke.
 - font podrazumevani font.
 - fg boja pisanja.
 - postcommand dodeljuje se naziv procedure koja se poziva kada se izbacuje meni.
 - relief podrazumevani 3D efekat je relief=RAISED.
 - image prikaz slike.
 - selectcolor boja selekcije.

Python, Menu

- `tearoff` normalno je tear-off element na 0-toj poziciji, tako da opcije počinju od broja 1, inače, ako ga nema opcije počinju od 0
- `title` promena tear-off naslova na dati string.

Metode su kao što sledi:

- `add_command (options)` dodaje stavku menija u meni.
- `add_radiobutton( options )` kreira radio button stavku menija.
- `add_checkbutton( options )` kreira check button stavku menija.
- `add_cascade(options)` kreira novi hijerarhijski meni asocijacijom datog menija roditeljskom meniju.
- `add_separator()` dodavanje separatorske linije meniju.
- `add( type, options )` dodavanje specifičnog tipa stavke meniju.
- `delete( startindex [, endindex ])` briše stavke menija u datom opsegu.
- `entryconfig( index, options )` modifikacija stavke menija određene datim indeksom i navedene opcije.
- `index(item)` vraća indeks date stavke.
- `insert_separator ( index )` ubacuje separator na poziciji datom indeksom.
- `invoke ( index )` poziv procedure asociirane sa izborom datog indeksa (checkboxbutton: alternira set i cleared, radiobutton: set).
- `type ( index )` tip izbora specificiranog indeksom:
"cascade", "checkboxbutton", "command", "radiobutton",
"separator".

Python, Menu

```
from tkinter import *
def donothing():
    tl= Toplevel(root)
    button = Button(tl, text="Do nothing button")
    button.pack()

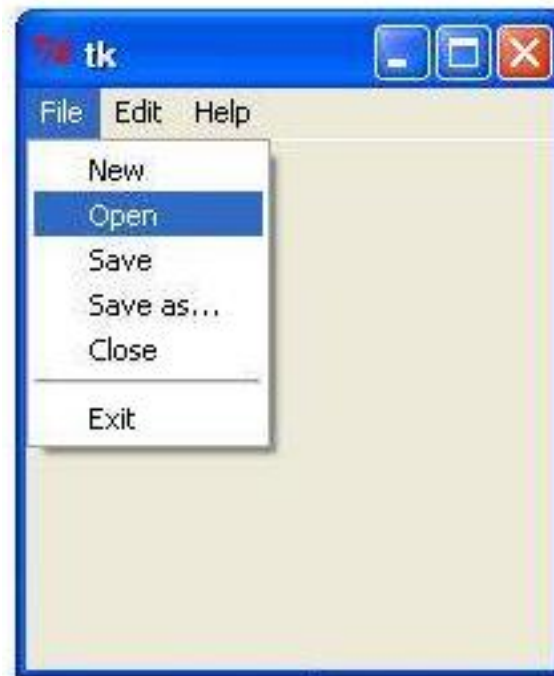
root = Tk()
menubar = Menu(root)
filemenu = Menu(menubar, tearoff=0)
menubar.add_cascade(label="File", menu=filemenu)
filemenu.add_command(label="New",          command=donothing)
filemenu.add_command(label="Open",         command=donothing)
filemenu.add_command(label="Save",         command=donothing)
filemenu.add_command(label="Save as...",  command=donothing)
filemenu.add_command(label="Close",       command=donothing)
filemenu.add_separator()
filemenu.add_command(label="Exit",        command=root.quit)
editmenu = Menu(menubar, tearoff=0)
menubar.add_cascade(label="Edit", menu=editmenu)
editmenu.add_command(label="Undo",        command=donothing)
editmenu.add_separator()
editmenu.add_command(label="Cut",         command=donothing)
editmenu.add_command(label="Copy",       command=donothing)
```

Python, Menu

```
editmenu.add_command(label="Paste",      command=donothing)  
editmenu.add_command(label="Delete",    command=donothing)  
editmenu.add_command(label="Select All", command=donothing)
```

```
helpmenu = Menu(menubar, tearoff=0)  
menubar.add_cascade(label="Help", menu=helpmenu)  
helpmenu.add_command(label="Help Index", command=donothing)  
helpmenu.add_command(label="About...",  command=donothing)
```

```
root.config(menu=menubar)  
root.mainloop()
```



Python, Scale

```
w = Scale ( master, option, ... )
```

Parametri:

- master: roditeljski prozor.
- options: date u CSV formi parova key=value.

Slede primeri opcija:

- activebackground boja pozadine kada je miš iznad widget-a.
- bg boja pozadine.
- bd širina 3D bordera oko klizača (podrazumevano 2 px).
- **command** procedura koja se poziva pri pomeranju klizača (argument je nova vrednost na skali).
- cursor tip kursora (arrow, dot, ...) iznad widget-a.
- **digits** dodeljivanje vrednosti kontrolnoj varijabli (IntVar, DoubleVar (float) ili StringVar). Za string, digits govori u koliko cifara se konvertuje vrednost na skali.
- font font koji se koristi.
- fg boja teksta.
- from_ vrednost početka opsega skale.

Python, Scale

- `label` labela (gore-levo za horizontalnu skalu, gore-desno za vertikalnu skalu, podrazumevano nema labele).
- `length` veličina widget-a (ovo je: x dimenzija ako je skala horizontalna, y dimenzija ako je skala vertikalna, podrazumevano je 100 px).
- `orient` orijentacija skale, podrazumevano je `orient=HORIZONTAL` (može biti `orient=VERTICAL`).
- `relief` dekorativni border oko labele (podrazumevano `FLAT`).
- `repeatdelay` kašnjenje ponavljanja pomeraja (podrazum. 300ms).
- `resolution` rezolucija skale (nps. skala je od -1.0 do 1.0, a rezolucija 0.5, vrednosti su: -1.0, -0.5, 0.0, 0.5 i 1.0).
- `showvalue` prikaz trenutne vrednosti (tekst je za horizontalnu postavljen iznad a za vertikalnu skalu levo do skale). Vrednost 0 znači da nema prikaza labele.
- `sliderlength` dužina klizača (podrazumevano 30 px).

Python, Scale

- state stanje widget-a (onemogućavanje: state=DISABLED).
- takefocus uzimanje fokusa preko tab-a. (takefocus=1, a za sprečavanje takefocus=0).
- tickinterval prikaz podela na skali (podrazumevano 0). Ako je opseg 0-1, onda su prikazi 0.0, 0.25, 0.50, 0.75 i 1.00 (za rezoluciju 0.25).
- to broj koji određuje kraj opsega skale.
- throughcolor boja trake.
- **variable** kontrolna varijabla skale (IntVar, DoubleVar ili StringVar). Za string se vrši konverzija vrednosti.
- width širina trake (podrazumevano 15 px).

Python, Scale

Metode:

- `get()` vraća tekuću vrednost skale.
- `set ( value )` postavlja vrednost skale.

```
from tkinter import *
def sel():
    selection = "Value = " + str(var.get())
    label.config(text = selection)
root = Tk()
var = DoubleVar()
scale = Scale( root, variable = var )
scale.pack(anchor=CENTER)
button = Button(root, text="Get Scale Value", command=sel)
button.pack(anchor=CENTER)
label = Label(root)
label.pack()
root.mainloop()
```



Python, Scrollbar

- `w = Scrollbar ( master, option, ... )`

Parametri:

- master: roditeljski prozor.
- options: date u CSV formi parova key=value.

Slede primeri opcija:

- `activebackground` boja klizača i strelica ispod miša.
- `bg` boja klizača i strelica kad miš nije iznad.
- `bd` širina 3D bordera (podrazumevano oko trake je nema , a oko strelica i klizača 2px).
- `command` procedura koja se zove kada se pomeri scrollbar.
- `cursor` tip kursora koji je iznad widget-a.
- `elementborderwidth` širina bordera klizača i strelica.

Python, Scrollbar

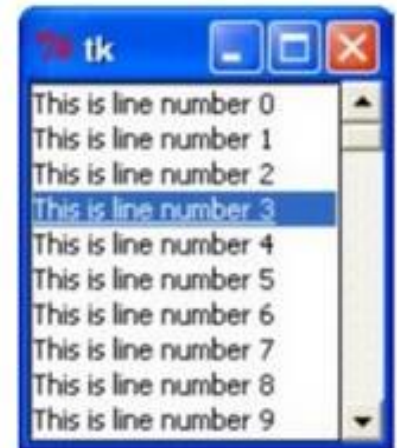
- `jump` opcija koja kontroliše šta se dešava kada korisnik pomera klizač. Ako je `jump=0` onda mali pomeraj klizača poziva callback proceduru, a ako je `jump=1` onda će se callback procedura pozvati kada korisnik otpusti dugme miša.
- `orient` orijentacija (`orient=HORIZONTAL` za horizontalni scrollbar, odnosno `orient=VERTICAL` za vertikalni scrollbar).
- `repeatdelay` odlaganje ponavljanja pomeraja kada je pritisnut taster 1 (podrazumevano 300ms).
- `repeatinterval` interval ponavljanja
- `takefocus` uzimanje fokusa preko tab-a (`takefocus=1`, a za sprečavanje `takefocus=0`).
- `throughcolor` boja trake.
- `width` širina scrollbar-a (za horizontalni to je y dimenzija, a za vertikalni to je x dimenzija, podrazumevano 16).

Python, Scrollbar

Metode:

- `get()` vraća poziciju klizača.
- `set ( first, last )` postavlja poziciju klizača.
 - Za konekciju nekog widget-a sa Scrollbar-om čiji je identifikator npr. scrollbar potrebno je parametru `xscrollcommand` ili `yscrollcommand` tog widget-a dodeliti vrednost `scrollbar.set`

```
from tkinter import *
root = Tk()
scrollbar = Scrollbar(root)
scrollbar.pack( side = RIGHT, fill=Y )
mylist = Listbox(root, yscrollcommand = scrollbar.set )
for line in range(100):
    mylist.insert(END, "This is line number " + str(line))
mylist.pack( side = LEFT, fill = BOTH )
scrollbar.config( command = mylist.yview )
mainloop()
```



Python, LabelFrame

```
w = LabelFrame( master, option, ... )
```

Parametri:

- master: roditeljski prozor.
- options: date u CSV formi parova key=value.

Slede primeri opcija:

- | | |
|---------------|--|
| • bg | boja pozadine. |
| • bd | veličina border-a (podrazumevano 2px). |
| • cursor | tip kursora kada je miš iznad komponenete. |
| • font | font teksta. |
| • width | širina widget-a. |
| • height | visina widget-a. |
| • labelAnchor | specificira gde se postavlja labela. |
| • relief | podrazumevano relief=FLAT. |
| • text | tekst koji se prikazuje. |

Python, LabelFrame

```
from tkinter import *  
root = Tk()  
labelframe = LabelFrame(root, text="This is a LabelFrame")  
labelframe.pack(fill="both", expand="yes")  
left = Label(labelframe, text="Inside the LabelFrame")  
left.pack()  
root.mainloop()
```



Python, Spinbox

```
w = Spinbox( master, option, ... )
```

Parametri:

- master: roditeljski prozor.
- options: date u CSV formi parova key=value.

Slede primeri opcija:

- activebackground aktivna boja pozadine (miš je iznad klizača ili strelica).
- bg boja pozadine.
- bd širina 3D bordera (podrazumevano 2px oko klizača i strelica)
- **command** procedura koja se poziva pri pomeranju spina.
- cursor tip kursora kada je iznad spinbox-a.
- disabledbackground boja pozadine onemogućenog spinbox-a.
- disabledforeground boja pisanja onemogućenog spinbox-a.
- fg boja teksta.
- font font koji se koristi.
- format format stringa.

Python, Spinbox

- `from_` minimalna vrednost spina.
- `justify` podrazumevano LEFT.
- `relief` podrazumevano SUNKEN.
- `repeatdelay` odlaganje ponavljanja u ms.
- `repeatinterval` interval ponavljanja u ms.
- `state` stanje: NORMAL, DISABLED ili "readonly".
- `textvariable` nema podrazumevanu vrednost.
- `to` maksimalna vrednost spina.
- `validate` validacioni mod (podrazumevano NONE).
- `validatecommand` validaciona callback procedura.
- `vcmd` kao i `validatecommand`.
- `values` n-torka validnih vrednosti spinbox-a (`values()`).
- `width` širina widget-a u karakterima (podrazumevano 20).
- `wrap` ako je true onda prikazuje vrednosti kružno.
- `xscrollcommand` veza sa horizontalnim scrollbar-om (ova opcija se postavlja metodom `set` odgovarajućeg scrollbar-a).

Metode

- `delete(startindex [,endindex])`
 - briše opseg karaktera ili karakter.
- `get(startindex [,endindex])`
 - vraća specifičan karakter ili opseg teksta.
- `identify(x, y)`
 - identifikuje widget element na datoj lokaciji.
- `index(index)`
 - vraća apsolutnu vrednost indeksa baziranog na datom indeksu.
- `insert(index [,string]...)`
 - umetanje stringa od datog indeksa.
- `invoke(element)`
 - poziv spinbox dugmeta (“buttonup” or “buttondown”).

Python, Spinbox

```
from tkinter import *
master = Tk()
def komanda():
    print ("trenutno je spinbox na: ",var.get())
var=StringVar()
w = Spinbox(master, from_=0, to=10, command=komanda, textvariable=var)
w.pack()
mainloop()
```



Python, Text

```
w = Text ( master, option, ... )
```

Parametri:

- master: roditeljski prozor.
- options: date u CSV formi parova key=value.

Slede primeri opcija:

- bg boja pozadine widget-a.
- bd širina border-a oko widget-a (podrazumevano 2px).
- cursor tip kursora iznad widget-a.
- exportselection eksport selektovanog teksta (0 ili 1).
- font font teksta widget-a.
- fg boja teksta widget-a.
- height visina widget-a u linijama (mereno prema tekućoj veličini fonta).

Python, Text

- `insertbackground` boja kursora u insert modu (podrazumevano black).
- `insertborderwidth` veličina 3D bordera kursora u insert modu (podraz. 0).
- `insertofftime` blinkanje isključenog kursora u insert modu (podrazumevano je isključen 300ms)
- `insertontime` blinkanje uključenog kursora u insert modu (podrazumevano je uključen 600ms)
- `insertwidth` širina kursora u insert modu.
- `padx` unuštašnja leva i desna margina (podraz. 1px).
- `pady` unutrašnja gornja i donja margina (podraz. 1px).
- `relief` 3D prikaz widget-a (podrazumevano relief=SUNKEN).
- `selectbackground` boja pozadine selektovanog teksta.
- `selectborderwidth` širina bordera oko selektovanog teksta.
- `spacing1` razmak iznad teksta (iznad prve linije, podrazum. 0).
- `spacing2` dodatni razmak između linija teksta (podraz. 0).
- `spacing3` razmak ispod teksta (podrazumevano 0).

Python, Text

- `state` stanje widget-a (NORMAL, DISABLED).
- `tabs` kontrola tab karaktera.
- `width` širina widget-a u karakterima mereno prema tekućem fontu.
- `wrap` prelom teksta na nivou reči ili karaktera (`wrap=WORD` vrši prelom do na reč, `wrap=CHAR` vrši prelom do na karakter).
- `xscrollcommand` postavljanje horizontalne skrolabilnosti widget-a.
- `yscrollcommand` postavljanje vertikalne skrolabilnosti widget-a.

Python, Text

Metode:

- `delete(startindex [,endindex])` briše dati opseg teksta.
- `get(startindex [,endindex])` vraća tekst specificiranog opsega.
- `insert(index [,string]...)` umetanje stringa od datog indeksa.
- `see(index)` vraća da li je tekst datog indeksa vidljiv.
- Text widget podržava 3 različite pomoćne strukture:
 - Marks markeri označavaju poziciju između 2 karaktera unutar datog teksta
 - Tabs tabovi se koriste za asocijaciju imenovanih regiona teksta
 - Indexes indeksi

Python, Text

Metode za markere:

- `index(mark)` vraća liniju i kolonu datog markera.
- `mark_gravity(mark [,gravity])` vraća gravitaciju datog markera (opciono drugi parametar postavlja gravitaciju).
- `mark_names()` vraća sve markere text widget-a.
- `mark_set(mark, index)` postavlja marker na dati indeks.
- `mark_unset(mark)` uklanja dati marker.

Metode za tagove:

- `tag_add(tagname, startindex[,endindex] ...)`
 - taguje poziciju startindeks-a ili opsega startindex-endindex.
- `tag_config`
 - konfigurisanje: `justify(center, left, or right)`, `tabs(kao svojstvo tabs)` i `underline(podvlačenje tagovanog teksta)`.
- `tag_delete(tagname)` : briše dati tag.
- `tag_remove(tagname [,startindex[,endindex]] ...)`
 - uklanjanje tag-a sa datog opsega ili pozicije bez brisanja definicije tag-a.

Python, Text

```
from tkinter import *  
root = Tk()  
text = Text(root)  
text.insert(INSERT, "Hello.....")  
text.insert(END, "Bye Bye.....")  
text.pack()  
  
text.tag_add("here", "1.0", "1.4")  
text.tag_add("start", "1.8", "1.13")  
text.tag_config("here", background="yellow", foreground="blue")  
text.tag_config("start", background="black", foreground="green")  
root.mainloop()
```

