

Óbudai Egyetem
Kandó Kálmán Villamosmérnöki Kar
Python
Tkinter kezelő widget-ek

Dr. Schuster György

2017. november 13.

Button widget

A button widget egy klasszikus kezelő gombot helyez el az aktuális konténeren.

Button widget

A button widget egy klasszikus kezelő gombot helyez el az aktuális konténeren.

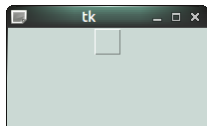
A **Button** widget létrehozása és paraméterezése:

Button widget

A button widget egy klasszikus kezelő gombot helyez el az aktuális konténeren.

A **Button** widget létrehozása és paraméterezése:

```
    ⋮  
b=Button (mw)  
    ⋮  
b.pack ()  
    ⋮
```



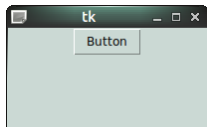
A **Button** widget létrehozása.

Button widget

A button widget egy klasszikus kezelő gombot helyez el az aktuális konténeren.

A **Button** widget létrehozása és paraméterezése:

```
    :  
b=Button (mw, text=' Button' )  
    :  
b.pack ()  
    :
```



A **Button** widget létrehozása.

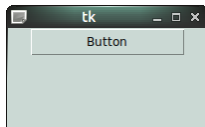
A felirat felrakása a gombra.

Button widget

A button widget egy klasszikus kezelő gombot helyez el az aktuális konténeren.

A **Button** widget létrehozása és paraméterezése:

```
    :  
b=Button(mw, text='Button', width=16)  
    :  
b.pack()  
    :
```



A **Button** widget létrehozása.

A felirat felrakása a gombra.

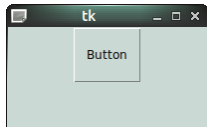
A gomb szélessége megadható, ekkor ezt karakter méretben értelmezi.

Button widget

A button widget egy klasszikus kezelő gombot helyez el az aktuális konténeren.

A **Button** widget létrehozása és paraméterezése:

```
    :  
b=Button(mw, text='Button', height=3)  
    :  
b.pack()  
    :
```



A **Button** widget létrehozása.

A felirat felrakása a gombra.

A gomb szélessége megadható, ekkor ezt karakter méretben értelmezi.

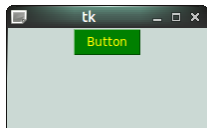
A gomb magassága is megadható, ekkor ezt karakter sor méretben értelmezi.

Button widget

A button widget egy klasszikus kezelő gombot helyez el az aktuális konténeren.

A **Button** widget létrehozása és paraméterezése:

```
    :  
b=Button(mw, text='Button', bg='green', fg='yellow')  
    :  
b.pack()  
    :
```



A Button widget létrehozása.

A felirat felrakása a gombra.

A gomb szélessége megadható, ekkor ezt karakter méretben értelmezi.

A gomb magassága is megadható, ekkor ezt karakter sor méretben értelmezi.

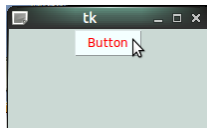
Az előtér és a háttér szín megadása. Ez a kombináció akkor érvényes, ha nincs a gomb felett a cursor.

Button widget

A button widget egy klasszikus kezelő gombot helyez el az aktuális konténeren.

A **Button** widget létrehozása és paraméterezése:

```
    :  
b=Button(mw, text='Button')  
b.config(activeforeground='red')  
b.pack()  
    :
```



A Button widget létrehozása.

A felirat felrakása a gombra.

A gomb szélessége megadható, ekkor ezt karakter méretben értelmezi.

A gomb magassága is megadható, ekkor ezt karakter sor méretben értelmezi.

Az előtér és a háttér szín megadása. Ez a kombináció akkor érvényes, ha nincs a gomb felett a cursor.

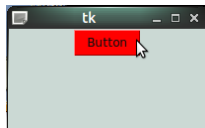
Az az előtérszín, amely akkor érvényes, ha a cursor a gomb felett van.

Button widget

A button widget egy klasszikus kezelő gombot helyez el az aktuális konténeren.

A **Button** widget létrehozása és paraméterezése:

```
    :  
b=Button(mw, text='Button')  
b.config(activebackground='red')  
b.pack()  
    :
```



A **Button** widget létrehozása.

A felirat felrakása a gombra.

A gomb szélessége megadható, ekkor ezt karakter méretben értelmezi.

A gomb magassága is megadható, ekkor ezt karakter sor méretben értelmezi.

Az előtér és a háttér szín megadása. Ez a kombináció akkor érvényes, ha nincs a gomb felett a cursor.

Az az előtérszín, amely akkor érvényes, ha a cursor a gomb felett van.

Az az háttérszín, amely akkor érvényes, ha a cursor a gomb felett van.

Button widget

A gomb eseményét egy úgynevezett callback függvény kezeli le.

Button widget

A gomb eseményét egy úgynevezett callback függvény kezeli le.

```
    :  
def fgv():  
    print('Callback')  
    :  
b=Button(mw, text='Button', command=fgv)  
b.pack()  
    :
```

Képernyő

Nyomjuk le a gombot!

A `command=` után megadjuk a callback függvény nevét.

Button widget

A gomb eseményét egy úgynevezett callback függvény kezeli le.

```
    :  
def fgv():  
    print('Callback')  
    :  
b=Button(mw, text='Button', command=fgv)  
b.pack()  
    :
```

Képernyő

Callback

Nyomjuk le a gombot!

A **command=** után megadjuk a callback függvény nevét.

A gomb lenyomása, majd elengedése esetén az **fgv** függvény meghívásra kerül.

Button widget

A gomb eseményét egy úgynevezett callback függvény kezeli le.

```
    :  
def fgv(a):  
    print(a)  
    :  
b=Button(mw, text='Button', command=lambda: fgv('Callback'))  
b.pack()  
    :
```

Képernyő

Nyomjuk le a
gombot megint!

A **command=** után megadjuk a callback függvény nevét.

A gomb lenyomása, majd elengedése esetén az **fgv** függvény meghívásra kerül.

A callback függvénynek át szeretnénk adni paramétert.

ekkor kell használni a **lambda: függvényt**.

Button widget

A gomb eseményét egy úgynevezett callback függvény kezeli le.

```

:
def fgv(a):
    print(a)
:
b=Button(mw, text='Button', command=lambda: fgv('Callback'))
b.pack()
:

```

Képernyő

Callback

Nyomjuk le a gombot/megint!

A **lambda** a funkcionális programozásból lett "átmentve".

És a szintaktikai szilárdsággal mivan?!

A **command=** után

A gomb lenyomása

A callback függvénynek az szeretnénk adni paraméterül.

ekkor kell használni a **lambda: függvényt**.

ra kerül.

Checkbutton widget

A `Checkbutton` widget egy klasszikus jelölő négyzetet valósít meg.

Checkbutton widget

A `Checkbutton` widget egy klasszikus jelölő négyzetet valósít meg.

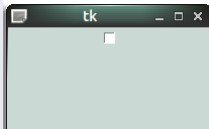
Használata és paraméterezése:

Checkbox widget

A `Checkbox` widget egy klasszikus jelölő négyzetet valósít meg.

Használata és paraméterezése:

```
    :  
a=IntVal()  
def fgv:  
    print(a)  
    :  
cb1=Checkbox(mw, variable=a, command=fgv)  
cb1.pack()  
    :  
    :
```



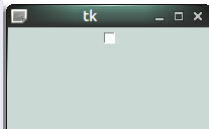
A widget létrehozása és elhelyezése.

Checkbox widget

A `Checkbox` widget egy klasszikus jelölő négyzetet valósít meg.

Használata és paraméterezése:

```
    :  
a=IntVal ()  
def fgv:  
    print (a)  
    :  
cb1=Checkbox (mw, variable=a, command=fgv)  
cb1.pack ()  
    :  
    :
```



A widget létrehozása és elhelyezése.

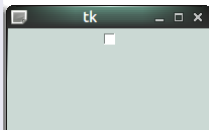
A változó, ahová a jelölőnégyzet állapota másolódik.

Checkbox widget

A `Checkbox` widget egy klasszikus jelölő négyzetet valósít meg.

Használata és paraméterezése:

```
⋮  
a=IntVal ()  
def fgv:  
    print (a)  
⋮  
cb1=Checkbox (mw, variable=a, command=fgv)  
cb1.pack ()  
⋮  
⋮
```



A widget létrehozása és elhelyezése.

A változó, ahová a jelölőnégyzet állapota másolódik.

Az eseménykezelő.

Checkbox widget

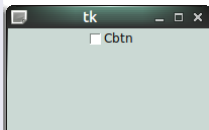
A `Checkbox` widget egy klasszikus jelölő négyzetet valósít meg.

Használata és paraméterezése:

```

:
a=IntVal()
def fgv:
    print(a)
:
cb1=Checkbox(mw, variable=a, command=fgv, text=' Cbtn' )
cb1.pack()
:
:

```



A widget létrehozása és elhelyezése.

A változó, ahová a jelölőnégyzet állapota másolódik.

Az eseménykezelő.

A jelölőnégyzet elnevezése.

Képernyő

Checkbox widget

A `Checkbox` widget egy klasszikus jelölő négyzetet valósít meg.

Használata és paraméterezése:

```

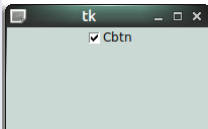
:
a=IntVal()
def fgv:
    print(a)
:
cb1=Checkbox(mw, variable=a, command=fgv, text=' Cbtn' )
cb1.pack()
:
:

```

A 'be' állapot default 1.

Képernyő

1



A widget létrehozása és elhelyezése.

A változó, ahová a jelölőnégyzet állapota másolódik.

Az eseménykezelő.

A jelölőnégyzet elnevezése.

Checkbox widget

A `Checkbox` widget egy klasszikus jelölő négyzetet valósít meg.

Használata és paraméterezése:

```

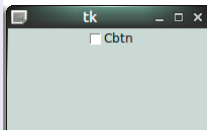
:
a=IntVal()
def fgv:
    print(a)
:
cb1=Checkbox(mw, variable=a, command=fgv, text=' Cbtn' )
cb1.pack()
:
:

```

A 'be' állapot default 1.
A 'ki' állapot default 0.

Képernyő

1
0



A widget létrehozása és elhelyezése.

A változó, ahová a jelölőnégyzet állapota másolódik.

Az eseménykezelő.

A jelölőnégyzet elnevezése.

Radiobutton widget

Klasszikus rádiógombot valósít meg.

Radiobutton widget

Klasszikus rádiógombot valósít meg.

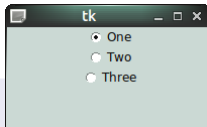
Létrehozása és paraméterezése:

Radiobutton widget

Klasszikus rádiógombot valósít meg.

Létrehozása és paraméterezése:

```
    :  
a=IntVar()  
    :  
def fgv():  
    print(a.get())  
    :  
rb1=Radiobutton(mw, variable=a, value=1, text=' One ', command=fgv)  
rb2=Radiobutton(mw, variable=a, value=2, text=' Two ', command=fgv)  
rb3=Radiobutton(mw, variable=a, value=3, text=' Three ', command=fgv)  
rb1.select()  
rb1.pack()  
rb2.pack()  
rb3.pack()  
    :
```



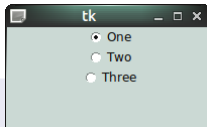
A rádiógombok létrehozása és elhelyezése.

Radiobutton widget

Klasszikus rádiógombot valósít meg.

Létrehozása és paraméterezése:

```
    :
a=IntVar()
    :
def fgv():
    print(a.get())
    :
rb1=Radiobutton(mw, variable=a, value=1, text=' One ', command=fgv)
rb2=Radiobutton(mw, variable=a, value=2, text=' Two ', command=fgv)
rb3=Radiobutton(mw, variable=a, value=3, text=' Three ', command=fgv)
rb1.select()
rb1.pack()
rb2.pack()
rb3.pack()
    :
```



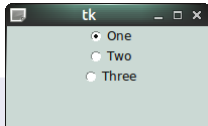
A rádiógombok létrehozása és elhelyezése.
Azonos nevű változó jelzi a csoportot.

Radiobutton widget

Klasszikus rádiógombot valósít meg.

Létrehozása és paraméterezése:

```
    :
a=IntVar()
    :
def fgv():
    print(a.get())
    :
rb1=Radiobutton(mw, variable=a, value=1, text='One', command=fgv)
rb2=Radiobutton(mw, variable=a, value=2, text='Two', command=fgv)
rb3=Radiobutton(mw, variable=a, value=3, text='Three', command=fgv)
rb1.select()
rb1.pack()
rb2.pack()
rb3.pack()
    :
```



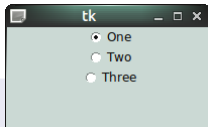
A rádiógombok létrehozása és elhelyezése.
Azonos nevű változó jelzi a csoportot.
Az eseményre visszaadott érték.

Radiobutton widget

Klasszikus rádiógombot valósít meg.

Létrehozása és paraméterezése:

```
    :  
a=IntVar()  
    :  
def fgv():  
    print(a.get())  
    :  
rb1=Radiobutton(mw, variable=a, value=1, text='One', command=fgv)  
rb2=Radiobutton(mw, variable=a, value=2, text='Two', command=fgv)  
rb3=Radiobutton(mw, variable=a, value=3, text='Three', command=fgv)  
rb1.select()  
rb1.pack()  
rb2.pack()  
rb3.pack()  
    :
```



A rádiógombok létrehozása és elhelyezése.

Azonos nevű változó jelzi a csoportot.

Az eseményre visszaadott érték.

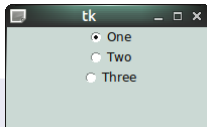
A gombok neve.

Radiobutton widget

Klasszikus rádiógombot valósít meg.

Létrehozása és paraméterezése:

```
    :
a=IntVar ()
    :
def fgv () :
    print (a.get ())
    :
rb1=Radiobutton (mw, variable=a, value=1, text=' One ' , command=fgv)
rb2=Radiobutton (mw, variable=a, value=2, text=' Two ' , command=fgv)
rb3=Radiobutton (mw, variable=a, value=3, text=' Three ' , command=fgv)
rb1.select ()
rb1.pack ()
rb2.pack ()
rb3.pack ()
    :
```



A rádiógombok létrehozása és elhelyezése.

Azonos nevű változó jelzi a csoportot.

Az eseményre visszaadott érték.

A gombok neve.

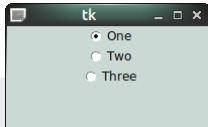
Az eseménykezelő.

Radiobutton widget

Klasszikus rádiógombot valósít meg.

Létrehozása és paraméterezése:

```
    :  
a=IntVar()  
    :  
def fgv():  
    print(a.get())  
    :  
rb1=Radiobutton(mw, variable=a, value=1, text=' One ', command=fgv)  
rb2=Radiobutton(mw, variable=a, value=2, text=' Two ', command=fgv)  
rb3=Radiobutton(mw, variable=a, value=3, text=' Three ', command=fgv)  
rb1.select()  
rb1.pack()  
rb2.pack()  
rb3.pack()  
    :
```



A rádiógombok létrehozása és elhelyezése.

Azonos nevű változó jelzi a csoportot.

Az eseményre visszaadott érték.

A gombok neve.

Az eseménykezelő.

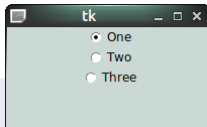
Az alapérték kiválasztása.

Radiobutton widget

Klasszikus rádiógombot valósít meg.

Létrehozása és paraméterezése:

```
    :  
a=IntVar()  
    :  
def fgv():  
    print(a.get())  
    :  
rb1=Radiobutton(mw, variable=a, value=1, text='One', command=fgv)  
rb2=Radiobutton(mw, variable=a, value=2, text='Two', command=fgv)  
rb3=Radiobutton(mw, variable=a, value=3, text='Three', command=fgv)  
rb1.select()  
rb1.pack()  
rb2.pack()  
rb3.pack()
```



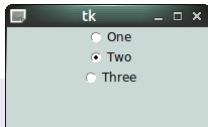
Képernyő

Radiobutton widget

Klasszikus rádiógombot valósít meg.

Létrehozása és paraméterezése:

```
    :  
a=IntVar()  
    :  
def fgv():  
    print(a.get())  
    :  
rb1=Radiobutton(mw, variable=a, value=1, text=' One ', command=fgv)  
rb2=Radiobutton(mw, variable=a, value=2, text=' Two ', command=fgv)  
rb3=Radiobutton(mw, variable=a, value=3, text=' Three ', command=fgv)  
rb1.select()  
rb1.pack()  
rb2.pack()  
rb3.pack()
```



Képernyő

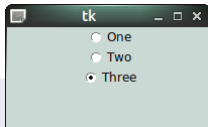
2

Radiobutton widget

Klasszikus rádiógombot valósít meg.

Létrehozása és paraméterezése:

```
    :  
a=IntVar ()  
    :  
def fgv () :  
    print (a.get ())  
    :  
rb1=Radiobutton (mw, variable=a, value=1, text=' One ' , command=fgv)  
rb2=Radiobutton (mw, variable=a, value=2, text=' Two ' , command=fgv)  
rb3=Radiobutton (mw, variable=a, value=3, text=' Three ' , command=fgv)  
rb1.select ()  
rb1.pack ()  
rb2.pack ()  
rb3.pack ()
```



Képernyő

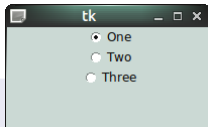
2
3

Radiobutton widget

Klasszikus rádiógombot valósít meg.

Létrehozása és paraméterezése:

```
    :  
a=IntVar()  
    :  
def fgv():  
    print(a.get())  
    :  
rb1=Radiobutton(mw, variable=a, value=1, text='One', command=fgv)  
rb2=Radiobutton(mw, variable=a, value=2, text='Two', command=fgv)  
rb3=Radiobutton(mw, variable=a, value=3, text='Three', command=fgv)  
rb1.select()  
rb1.pack()  
rb2.pack()  
rb3.pack()
```



Képernyő

2
3
1

Scale widget

A `Scale` widget egy csúszkát tesz a kérdéses konténerre.

Scale widget

A `Scale` widget egy csúszkát tesz a kérdéses konténerre.

A `Scale` létrehozása és paraméterezése:

Scale widget

A `Scale` widget egy csúszkát tesz a kérdéses konténerre.

A `Scale` létrehozása és paraméterezése:

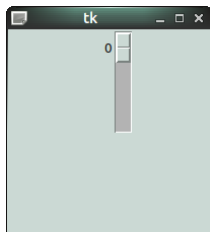
```
    :  
a=0  
def fgv(a):  
    print(a)  
    :  
sc=Scale(mw, variable=a, command=fgv)  
sc.pack()  
    :
```

A widget létrehozása és elhelyezése.

Az `ős` widget.

Az állást visszaadó változó.

A callback függvény.



Képernyő

0

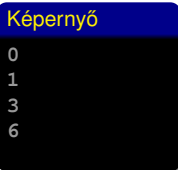
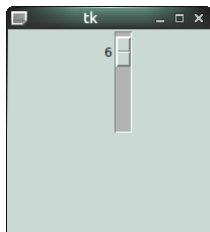
Scale widget

A `Scale` widget egy csúszkát tesz a kérdéses konténerre.

A `Scale` létrehozása és paraméterezése:

```
    :  
a=0  
def fgv(a):  
    print(a)  
    :  
sc=Scale(mw, variable=a, command=fgv)  
sc.pack()  
    :
```

Mozdítsuk meg a csúszkát!



Scale widget

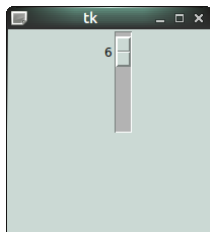
A `Scale` widget egy csúszkát tesz a kérdéses konténerre.

A `Scale` létrehozása és paraméterezése:

```

:
a=0
def fgv(a):
    print(a)
:
sc=Scale(mw, variable=a, command=fgv)
sc.pack()
:

```



Mozdítsuk meg a csúszkát!

Képernyő

Sajnos a sok paraméter nem fér ki az ablakban, ezért a

`mw, variable=a, command=fgv`
paramétereket a továbbiakban egy ...
szimbólummal jelöljük.

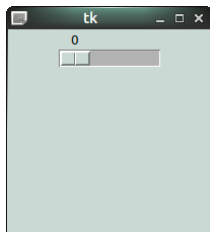
Scale widget

A **Scale** widget egy csúszkát tesz a kérdéses konténerre.

A **Scale** létrehozása és paraméterezése:

```
    :  
a=0  
def fgv(a):  
    print(a)  
    :  
sc=Scale(..., orient=HORIZONTAL)  
sc.pack()  
    :
```

A widget iránya (**orient=VERTICAL** a default).



Scale widget

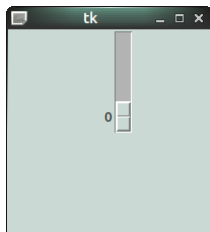
A **Scale** widget egy csúszkát tesz a kérdéses konténerre.

A **Scale** létrehozása és paraméterezése:

```
    :  
a=0  
def fgv(a):  
    print(a)  
    :  
sc=Scale(..., from_=100, to=0)  
sc.pack()  
    :
```

A widget iránya (**orient=VERTICAL** a default).

A csúszka értelmezési tartománya, honnan, hova.



Scale widget

A **Scale** widget egy csúszkát tesz a kérdéses konténerre.

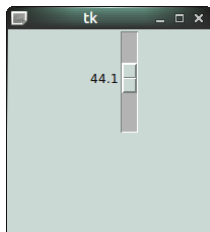
A **Scale** létrehozása és paraméterezése:

```
    :  
a=0  
def fgv(a):  
    print(a)  
    :  
sc=Scale(..., resolution=0.1)  
sc.pack()  
    :
```

A widget iránya (**orient=VERTICAL** a default).

A csúszka értelmezési tartománya, honnan, hova.

A felbontás, (default: 1).



Scale widget

A **Scale** widget egy csúszkát tesz a kérdéses konténerre.

A **Scale** létrehozása és paraméterezése:

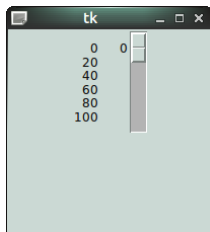
```
    :  
a=0  
def fgv(a):  
    print(a)  
    :  
sc=Scale(..., tickinterval=20)  
sc.pack()  
    :
```

A widget iránya (**orient=VERTICAL** a default).

A csúszka értelmezési tartománya, honnan, hova.

A felbontás, (default: 1).

Felskálázza a widget-et a megadott lépésközzel.



Scale widget

A **Scale** widget egy csúszkát tesz a kérdéses konténerre.

A **Scale** létrehozása és paraméterezése:

```
    :  
a=0  
def fgv(a):  
    print(a)  
    :  
sc=Scale(..., tickinterval=20)  
sc.pack()  
    :
```

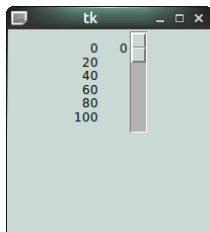
A widget iránya (**orient=VERTICAL** a default).

A csúszka értelmezési tartománya, honnan, hova.

A felbontás, (default: 1).

Felskálázza a widget-et a megadott lépésközzel.

A **Scale** widget függvényei:

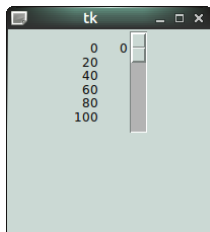


Scale widget

A **Scale** widget egy csúszkát tesz a kérdéses konténerre.

A **Scale** létrehozása és paraméterezése:

```
    :  
a=0  
def fgv(a):  
    print(a)  
    :  
sc=Scale(..., tickinterval=20)  
sc.pack()  
    :
```



A widget iránya (**orient=VERTICAL** a default).

A csúszka értelmezési tartománya, honnan, hova.

A felbontás, (default: 1).

Felskálázza a widget-et a megadott lépésközzel.

A **Scale** widget függvényei:

get a **Scale** widget aktuális értékét adja vissza,

Scale widget

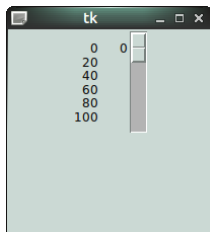
A **Scale** widget egy csúszkát tesz a kérdéses konténerre.

A **Scale** létrehozása és paraméterezése:

```

    :
a=0
def fgv(a):
    print(a)
    :
sc=Scale(..., tickinterval=20)
sc.pack()
    :

```



A widget iránya (**orient=VERTICAL** a default).

A csúszka értelmezési tartománya, honnan, hova.

A felbontás, (default: 1).

Felskálázza a widget-et a megadott lépésközzel.

A **Scale** widget függvényei:

get a **Scale** widget aktuális értékét adja vissza,

set beállítja a **Scale** widget aktuális értékét.

Label widget

Egy feliratot tesz ki a kiválasztott konténerre.

Label widget

Egy feliratot tesz ki a kiválasztott konténerre.

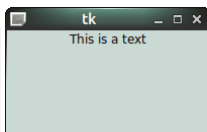
Elhelyezése és paraméterezése:

Label widget

Egy feliratot tesz ki a kiválasztott konténerre.

Elhelyezése és paraméterezése:

```
    :  
lb=Label(mw,text='This is a text')  
lb.pack()  
    :  
    :
```



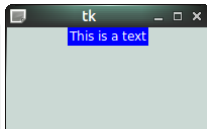
A widget létrehozása és elhelyezése.

Label widget

Egy feliratot tesz ki a kiválasztott konténerre.

Elhelyezése és paraméterezése:

```
⋮  
lb=Label(mw,text='This is a text',fg=white,bg='blue')  
lb.pack()  
⋮  
⋮
```



A widget létrehozása és elhelyezése.

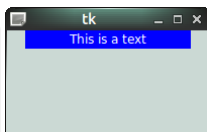
Az előtér és a háttér színezése.

Label widget

Egy feliratot tesz ki a kiválasztott konténerre.

Elhelyezése és paraméterezése:

```
    :  
lb=Label(mw,text='This is a text',fg=white,bg='blue')  
lb.config(width=20)  
lb.pack()  
    :
```



A widget létrehozása és elhelyezése.

Az előtér és a háttér színeztése.

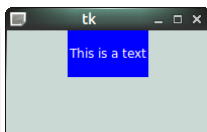
A szélesség beállítása (karakterben értelmezve).

Label widget

Egy feliratot tesz ki a kiválasztott konténerre.

Elhelyezése és paraméterezése:

```
    :  
lb=Label(mw,text='This is a text',fg=white,bg='blue')  
lb.config(height=3)  
lb.pack()  
    :
```



A widget létrehozása és elhelyezése.

Az előtér és a háttér színezése.

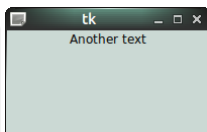
A szélesség beállítása (karakterben értelmezve).

A magasság beállítása (sorban értelmezve).

Label widget

Egy feliratot tesz ki a kiválasztott konténerre.
Elhelyezése és paraméterezése:

```
a=StringVar()  
lb=Label(mw, textvariable=a)  
a.set("Another text")  
lb.pack()  
:
```

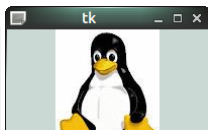


A widget létrehozása és elhelyezése.
Az előtér és a háttér színezése.
A szélesség beállítása (karakterben értelmezve).
A magasság beállítása (sorban értelmezve).
A szöveg változó használata.

Label widget

Egy feliratot tesz ki a kiválasztott konténerre.
Elhelyezése és paraméterezése:

```
    :  
    tux=PhotoImage (file=tux.gif)  
    lb=Label (mw, image=tux)  
    lb.pack ()  
    :
```



A widget létrehozása és elhelyezése.
Az előtér és a háttér színezése.
A szélesség beállítása (karakterben értelmezve).
A magasság beállítása (sorban értelmezve).
A szöveg változó használata.
Egy gif kép kirakása.

Menu widget

A **Menu** widget elhelyezése és paraméterezése:

Menu widget

A **Menu** widget elhelyezése és paraméterezése:

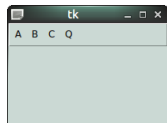
```
    :  
def fgv0  
    pass  
    :  
def fgvn  
    pass  
    :
```

Callback függvények igény szerint.

Menu widget

A **Menu** widget elhelyezése és paraméterezése:

```
    :  
mb=Menu (mw)  
mb.add_command(label='A', command=fgv1)  
mb.add_command(label='B', command=fgv2)  
mb.add_command(label='C', command=fgv3)  
mb.add_command(label='Q', command=mw.quit)  
mw.config(menu=mb)  
    :
```



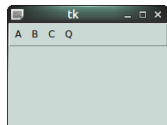
Callback függvények igény szerint.

A menü létrehozása.

Menu widget

A **Menu** widget elhelyezése és paraméterezése:

```
    :  
mb=Menu (mw)  
mb.add_command(label='A', command=fgv1)  
mb.add_command(label='B', command=fgv2)  
mb.add_command(label='C', command=fgv3)  
mb.add_command(label='Q', command=mw.quit)  
mw.config(menu=mb)  
    :
```



Callback függvények igény szerint.

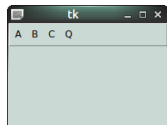
A menü létrehozása.

A menü "kirakása" az ablakra.

Menu widget

A **Menu** widget elhelyezése és paraméterezése:

```
    :  
mb=Menu (mw)  
mb.add_command(label='A', command=fgv1)  
mb.add_command(label='B', command=fgv2)  
mb.add_command(label='C', command=fgv3)  
mb.add_command(label='Q', command=mw.quit)  
mw.config(menu=mb)  
    :
```



Callback függvények igény szerint.

A menü létrehozása.

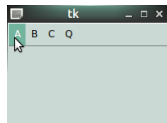
A menü "kirakása" az ablakra.

A menüpontok elhelyezése.

Menu widget

A **Menu** widget elhelyezése és paraméterezése:

```
    :  
mb=Menu (mw)  
mb.add_command(label='A', command=fgv1)  
mb.add_command(label='B', command=fgv2)  
mb.add_command(label='C', command=fgv3)  
mb.add_command(label='Q', command=mw.quit)  
mw.config(menu=mb)  
    :
```



Callback függvények igény szerint.

A menü létrehozása.

A menü "kirakása" az ablakra.

A menüpontok elhelyezése.

Kiválasztott menüpont. A meghívandó callback függvény.

Menu widget

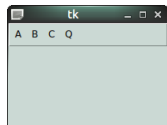
A **Menu** widget lehetővé teszi legördülő menük létrehozását.

Menu widget

A **Menu** widget lehetővé teszi legördülő menük létrehozását.

```
    :  
mb=Menu (mw)  
am=Menu (mb, tearoff=0)  
    :  
am.add_command (label='A1', command=fgv)  
am.add_command (label='A2', command=fgv)  
mb.add_cascade (label='A', menu=am)  
    :  
mw.config (menu=mb)  
    :  
    :
```

Az főmenü létrehozása.



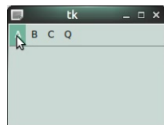
Menu widget

A **Menu** widget lehetővé teszi legördülő menük létrehozását.

```
    :
mb=Menu (mw)
am=Menu (mb, tearoff=0)
    :
am.add_command (label=' A1 ', command=fgv)
am.add_command (label=' A2 ', command=fgv)
mb.add_cascade (label=' A ', menu=am)
    :
mw.config (menu=mb)
    :
    :
```

Az főmenü létrehozása.

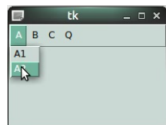
A legördülő menü létrehozása.



Menu widget

A **Menu** widget lehetővé teszi legördülő menük létrehozását.

```
    :  
mb=Menu (mw)  
am=Menu (mb, tearoff=0)  
    :  
am.add_command (label=' A1 ', command=fgv)  
am.add_command (label=' A2 ', command=fgv)  
mb.add_cascade (label=' A ', menu=am)  
    :  
mw.config (menu=mb)  
    :  
    :
```



Az főmenü létrehozása.

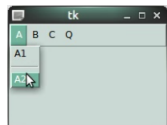
A legördülő menü létrehozása.

Menüpontok elhelyezése.

Menu widget

A **Menu** widget lehetővé teszi legördülő menük létrehozását.

```
    :
mb=Menu (mw)
am=Menu (mb, tearoff=0)
    :
am.add_command (label=' A1 ', command=fgv)
am.add_command (label=' A2 ', command=fgv)
am.add_separator ()
mb.add_cascade (label=' A ', menu=am)
    :
mw.config (menu=mb)
    :
```



Az főmenü létrehozása.

A legördülő menü létrehozása.

Menüpontok elhelyezése.

Elválasztó vonal berakása.

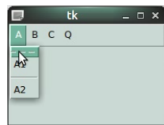
Menu widget

A **Menu** widget lehetővé teszi legördülő menük létrehozását.

```

:
mb=Menu (mw)
am=Menu (mb, tearoff=1)
:
am.add_command (label=' A1 ', command=fgv)
am.add_command (label=' A2 ', command=fgv)
am.add_separator ()
mb.add_cascade (label=' A ', menu=am)
:
mw.config (menu=mb)
:

```



Az főmenü létrehozása.

A legördülő menü létrehozása.

Menüpontok elhelyezése.

Elválasztó vonal berakása.

A levágható menü beállítása ...

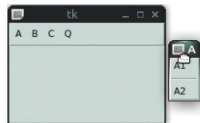
Menu widget

A **Menu** widget lehetővé teszi legördülő menük létrehozását.

```

:
mb=Menu (mw)
am=Menu (mb, tearoff=1)
:
am.add_command (label=' A1 ', command=fgv)
am.add_command (label=' A2 ', command=fgv)
am.add_separator ()
mb.add_cascade (label=' A ', menu=am)
:
mw.config (menu=mb)
:

```



Az főmenü létrehozása.

A legördülő menü létrehozása.

Menüpontok elhelyezése.

Elválasztó vonal berakása.

A levágható menü beállítása ...

...és el is húzzuk.

OptionMenu widget

Egy legördülő listát tesz lehetővé.

OptionMenu widget

Egy legördülő listát tesz lehetővé.
Elhelyezése és paraméterezése:

OptionMenu widget

Egy legördülő listát tesz lehetővé.

Elhelyezése és paraméterezése:

```
    :  
def fgv():  
    print(var.get())  
    :  
var=StringVar()  
om=OptionMenu(mw,var,"aaa","bbb","ccc")  
om.pack()  
bt=Button(mw,command=fgv)  
bt.pack()  
    :
```

Esemény a kiolvasáshoz.



OptionMenu widget

Egy legördülő listát tesz lehetővé.

Elhelyezése és paraméterezése:

```
⋮  
var=StringVar()  
om=OptionMenu(mw, var, "aaa", "bbb", "ccc")  
om.pack()  
⋮
```



Esemény a kiolvasáshoz.

A visszatott változó.

OptionMenu widget

Egy legördülő listát tesz lehetővé.

Elhelyezése és paraméterezése:

```
    :  
var=StringVar ()  
om=OptionMenu (mw, var, "aaa", "bbb", "ccc")  
om.pack ()  
    :
```



Esemény a kiolvasáshoz.

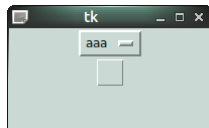
A visszatott változó.

A widget létrehozása és elhelyezése.

OptionMenu widget

Egy legördülő listát tesz lehetővé.
Elhelyezése és paraméterezése:

```
⋮  
var=StringVar()  
om=OptionMenu(mw, var, "aaa", "bbb", "ccc")  
om.pack()  
⋮
```

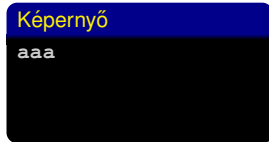


Esemény a kiolvasáshoz.

A visszatott változó.

A widget létrehozása és elhelyezése.

Az első "menüpont".



OptionMenu widget

Egy legördülő listát tesz lehetővé.
Elhelyezése és paraméterezése:

```
    :  
var=StringVar ()  
om=OptionMenu (mw, var, "aaa", "bbb", "ccc")  
om.pack ()  
    :
```



Esemény a kiolvasáshoz.

A visszatott változó.

A widget létrehozása és elhelyezése.

Az első "menüpont".

A második "menüpont".

Képernyő

aaa

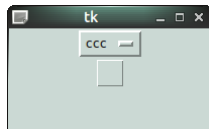
bbb

OptionMenu widget

Egy legördülő listát tesz lehetővé.

Elhelyezése és paraméterezése:

```
⋮  
var=StringVar()  
om=OptionMenu(mw,var,"aaa","bbb","ccc")  
om.pack()  
⋮
```



Esemény a kiolvasáshoz.

A visszatott változó.

A widget létrehozása és elhelyezése.

Az első "menüpont".

A második "menüpont".

A harmadik "menüpont".

Képernyő

aaa

bbb

ccc