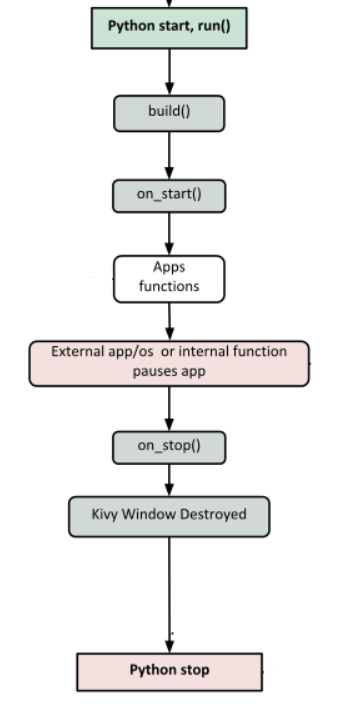
Kivy: egy nyílt forráskódú Python szoftver könyvtár Grafikus User Interfészek fejlesztésére (GUI).

A Kivy alkalmazások számos oprendszeren futtathatók, úgymint Windows, Linux, macOS, Android, IOS.

Kivy install:

cmd $ python -m pip install kivy

Kivy alkalmazás életciklusa:



A belépési pont az alkalmazásba a run() metódus.

Hogyan épül fel a kód?

A kivy alkalmazások készítéséhez:

- importálni kell a kivy.app modulból az App osztályt:

from kivy.app import App

Az App osztály az alap műveleteket tartalmazza, amelyek szükségesek GUI alkalmazás készítéséhez Kivyben, pl.: eseményhurok kezelése.

-származtatni kell a saját osztályunkat az App osztályból:

class MainApp(App):

def build(self):

A build() metódus szolgál a UI létrehozására, ezért a származtatott App osztály build() metódusát, mely egyszer hívódik meg, felülírjuk a saját osztályunkban. Az App osztály többi metódusa öröklődik, (azokat nem írtuk felül, tehát nem változtak meg), azokat meg tudjuk hívni.

A build metódusban elkészítünk és visszaadunk egy widget-et, vagy layout-ot, mely a gyökér elem lesz és hozzáadja az ablakhoz.

A build() metódus a belépési pont, amikor valami a képernyőre íródik.

return Label(text=”Hello World”)

Jelen esetben egy címke, mely a „Hello world” szöveget tartalmazza.

widget: a GUI elem, ami megjelenít információt, vagy valamilyen funkciót lát el. Ezek az elemek teszik lehetővé a felhasználónak, hogy kapcsolatba lépjenek a programmal.

Widget osztály: alap osztály, mely szükséges a widgetek létrehozásához.

Label widget: szöveg megjelenítésére ablakokon vagy dialógusban

TextInput widget: szerkeszthető szöveges mező

Button widget: nyomógomb. Esemény történik, amikor a felhasználó megnyomja.

CheckBox widget: kétállapotú nyomógomb, mely bejelölhető vagy törölhető

Image widget: kép megjelenítésére

ProgressBar widget: folyamatok állapotának megjelenítésére

Dropdown widget: legördülő lista létrehozására

Ha a gyökér widget-hez újabb widgetet akarunk hozzáadni gyerekelemeként, akkor az

**add\_widget**(widget, index=0, canvas=None)

metódussal tehetjük meg. (A widgetek Kivyben fa struktúrába vannak szervezve).

widget: a widget megnevezése, melyet a child elem listához adunk.

index: integer, alapból 0-a. A listában a beinsertált widget indexe.

A widget-ek kitöltik a rendelkezésre álló helyet területet, így feleslegesen nagy méretűek lehetnek. (Felesleges, hogy a nyomógomb kitöltse az egész ablakot).

A size\_hint: ReferenceListProperty(hivatkozás) a size\_hint\_x és size\_hint\_y, a widget x és y koordináták szerinti méretére. [0,1] közötti értéket vár. A default (1,1). Relatív widget méret.

Pozíció:

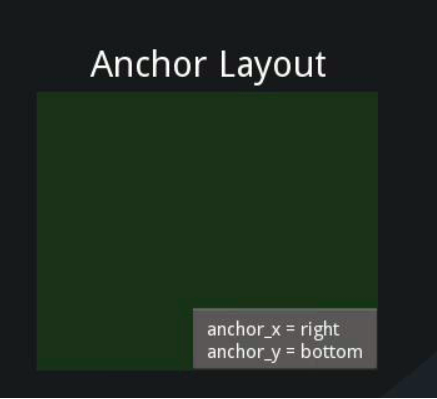
Abszolút: pos. pl.: pos=(200,200)

Relatív: pos\_hint={’x’:.3,’y’:.6}

A layout biztosítja, hogy a widgetek meghatározott elrendezés szerint jelenjenek meg az alkalmazás ablakában.

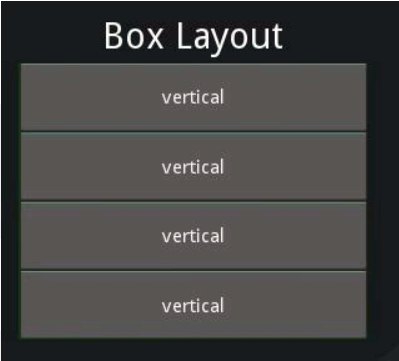
Layout: a widgetek elrendezése valamilyen struktúra szerint az alkalmazás ablakában.

AnchorLayout: A widget rögzíthető „felül”, „alul”, „balra”, „jobbra” vagy „középre”.



BoxLayout: A widgetek (gyerek elemek) egymás után vannak elrendezve, „függőleges” vagy „vízszintes” tájolásban.





FloatLayout: A widgetek elhelyezkedése lényegében korlátlan.



RelativeLayout: A gyerek widgetek relative módon vannak pozícionálva a layouthoz.

GridLayout: A widgetek gridbe, hálóba vannak elrendezve, melyet sorok és oszlopok definiálnak

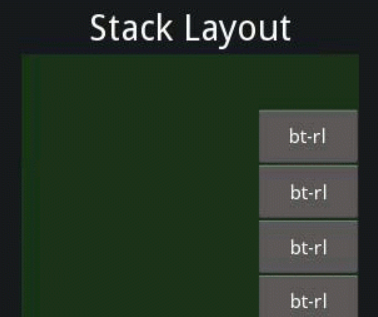


PageLayout: Egyszerű többoldalas elrendezések létrehozására. Az egyik oldalról a lapozás

a másikra a szegélyek használatával történik.

ScatterLayout: A widgetek a RelativeLayouthoz hasonlóan helyezkednek el, de lefordíthatók, elforgathatók és méretezhetők.

StackLayout: A widgetek balról jobbra, majd fentről lefelé (vagy fordítva) vannak egymásra rakva.



Amikor egy widgetet adunk a layouthoz, akkor a widget méretét és pozícióját határozzuk meg a layout típusától függően.

A widgeteket és layout-okat a kivy.uix modul: tartalmazza.

1. gyökér elem létrehozása
2. widget definiálása
3. widget hozzáadása a gyökérhez, vagy valamelyik widget listához. A widgeteket listába szervezem.

layout = BoxLayout**(**padding=10**)**

button = Button**(**text='My first button'**)**

layout.add\_widget**(**button**)**

Pl.:

from kivy.uix.label import Label

Végül, a MainApp osztályból egy hivatkozást hozunk létre és a származtatott run metódust meghívjuk.

A run metódus elindítja az alkalmazást és futtatja a fő hurkot.

https://kivy.org/doc/stable/api-kivy.uix.html#module-kivy.uix

GUI\_1.txt

Label widget:

from kivy.app import App  
from kivy.uix.label import Label  
  
class MainApp(App):  
 def build(self):  
 self.title="Elso"  
 return Label(text="Jó napot!")  
  
MainApp().run()

**Layout**

* Grid: GUI\_2\_grid.py

Hálóban (gridben) rendezi el a címkéket, melybe beleírja a koordinátákat.

from kivy.app import App  
from kivy.uix.label import Label  
from kivy.uix.gridlayout import GridLayout  
  
ROWS=COLS=2  
  
class MyApp(App):  
 def build(self):  
 layout=GridLayout(rows=ROWS,cols=COLS)  
  
 for i in range(ROWS):  
 for j in range(COLS):  
 concat=str(i)+","+str(j)  
 layout.add\_widget(Label(text=concat))  
  
 return layout  
  
MyApp().run()

* BoxLayout: GUI\_2\_box.py

Boxban rendezi el a címkéket, azokba írja bele a dátumot és magyarul és angolul a hét napját.

from kivy.app import App  
from kivy.uix.label import Label  
from kivy.uix.boxlayout import BoxLayout  
import time  
  
class MyDate():  
 def showdate(self):  
 datum=time.localtime()  
 weekdays\_hun=("Hétfő","Kedd","Szerda","Csütörtök","Péntek","Szombat","Vasárnap")  
 weekdays\_eng=("Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday")  
 datumstr=[str(datum[0])+'.'+str(datum[1])+'.'+str(datum[2])]  
 datumstr+=[weekdays\_hun[datum[6]]]  
 datumstr+=[weekdays\_eng[datum[6]]]  
 return (datumstr)  
  
class MyApp(App):  
 def build(self):  
 layout=BoxLayout(orientation='vertical')  
  
 mydatepl=MyDate()  
  
 layout.add\_widget(Label(text=mydatepl.showdate()[0]))  
 layout.add\_widget(Label(text=mydatepl.showdate()[1]))  
 layout.add\_widget(Label(text=mydatepl.showdate()[2]))  
  
 return layout  
  
MyApp().run()

Feladat: Írjon grafikus programot, melyben az órarendjét írja ki. GUI\_2\_grid\_timetable.py

Feladat: Írjon Naptár grafikus programot, mely bekéri a felhasználótól a hónap sorszámát, kiírja a sorszámnak megfelelő hónap nevet és rácsban annyi sorszámot, ahány nap van az adott hónapban.