from kivy.app import App

from kivy.uix.label import Label

from kivy.clock import Clock

from kivy.graphics import Color, Rectangle

*class* TestApp(*App*):

    @staticmethod

*def* get\_x(*label*, *ref\_x*):

        """ Return the x value of the ref/anchor relative to the canvas """

        return *label*.center\_x - *label*.texture\_size[0] \* 0.5 + *ref\_x*

    @staticmethod

*def* get\_y(*label*, *ref\_y*):

        """ Return the y value of the ref/anchor relative to the canvas """

        # Note the inversion of direction, as y values start at the top of

        # the texture and increase downwards

        return *label*.center\_y + *label*.texture\_size[1] \* 0.5 - *ref\_y*

*def* show\_marks(*self*, *label*):

        # Indicate the position of the anchors with a red top marker

        for name, anc in *label*.anchors.items():

            with *label*.canvas:

                Color(1, 0, 0)

                Rectangle(*pos*=(*self*.get\_x(*label*, anc[0]),

*self*.get\_y(*label*, anc[1])),

*size*=(3, 3))

        # Draw a green surround around the refs. Note the sizes y inversion

        for name, boxes in *label*.refs.items():

            for box in boxes:

                with *label*.canvas:

                    Color(0, 1, 0, 0.25)

                    Rectangle(*pos*=(*self*.get\_x(*label*, box[0]),

*self*.get\_y(*label*, box[1])),

*size*=(box[2] - box[0],

                                    box[1] - box[3]))

*def* build(*self*):

        label = Label(

*text*='[anchor=a]a\nChars [anchor=b]b\n[ref=myref]ref[/ref]',

*markup*=True)

        Clock.schedule\_once(*lambda* *dt*: *self*.show\_marks(label), 1)

        return label

TestApp().run()

=====================================================================================

import kivy

from docutils.nodes import image

Module: kivy.uix.image

Wimg: object = image(*source*='my-logo.JPG')

=====================================================================================

from kivy.core.image import Image as CoreImage

im = CoreImage("image.png")

=====================================================================================

import kivy

kivy.require('1.0.6') # replace with your current kivy version !

from kivy.app import App

from kivy.uix.label import Label

*class* MyApp(*App*):

*def* build(*self*):

        return Label(*text*='Hello world')

if \_\_name\_\_ == '\_\_main\_\_':

    MyApp().run()

=====================================================================================

# Program to explain how to add image in kivy

# import kivy module

import kivy

# base Class of your App inherits from the App class.

# app:always refers to the instance of your application

from kivy.app import App

# this restrict the kivy version i.e

# below this kivy version you cannot

# use the app or software

kivy.require('1.9.0')

# The Image widget is used to display an image

# this module contain all features of images

from kivy.uix.image import Image

# creating the App class

*class* MyApp(*App*):

    # defining build()

*def* build(*self*):

        # return image

        return Image(*source*='img2.jpg')

# run the App

MyApp().run()

=====================================================================================

# Program to Show how to use images in kivy

# import kivy module

import kivy

# base Class of your App inherits from the App class.

# app:always refers to the instance of your application

from kivy.app import App

# this restrict the kivy version i.e

# below this kivy version you cannot

# use the app or software

kivy.require('1.9.0')

# The Image widget is used to display an image

# this module contain all features of images

from kivy.uix.image import Image

# The Widget class is the base class required for creating Widgets

from kivy.uix.widget import Widget

# to change the kivy default settings we use this module config

from kivy.config import Config

# 0 being off 1 being on as in true / false

# you can use 0 or 1 && True or False

Config.set('graphics', 'resizable', True)

# creating the App class

*class* MyApp(*App*):

    # defining build()

*def* build(*self*):

        # loading image

*self*.img = Image(*source* ='download.jpg')

        # By default, the image is centered and fits

        # inside the widget bounding box.

        # If you don’t want that,

        # you can set allow\_stretch to

        # True and keep\_ratio to False.

*self*.img.allow\_stretch = True

*self*.img.keep\_ratio = False

        # Providing Size to the image

        # it varies from 0 to 1

*self*.img.size\_hint\_x = 1

*self*.img.size\_hint\_y = 1

        # Position set

*self*.img.pos = (200, 100)

        # Opacity adjust the fadeness of the image if

        # 0 then it is complete black

        # 1 then original

        # it varies from 0 to 1

*self*.img.opacity = 1

        # adding image to widget

        s = Widget()

        s.add\_widget(*self*.img)

        # return widget

        return s

# run the app

MyApp().run()

=====================================================================================

# Simple program to show how we add AsyncImage in kivy App

# import kivy module

import kivy

# base Class of your App inherits from the App class.

# app:always refers to the instance of your application

from kivy.app import App

# this restrict the kivy version i.e

# below this kivy version you cannot

# use the app or software

kivy.require('1.9.0')

# The Image widget is used to display an image

# this module contains all features of images

from kivy.uix.image import AsyncImage

# creating the App class

*class* MyApp(*App*):

    # defining build()

*def* build(*self*):

        # return image

        return AsyncImage(*source*='http://kivy.org/logos/kivy-logo-black-64.png')

# run the App

MyApp().run()

=====================================================================================

from kivy.app import App

from kivy.uix.button import Button

*class* HelloApp(*App*):

*def* build(*self*):

    return Button(*text*='Hello Berlin')

HelloApp().run()

=====================================================================================

import kivy

from kivy.app import App

from kivy.uix.widget import Widget

*class* PongGame(*Widget*):

    pass

*class* PongApp(*App*):

*def* build(*self*):

        return PongGame()

if \_\_name\_\_ == '\_\_main\_\_':

 PongApp().run()

=====================================================================================

from kivy . app import App

from kivy . config import Config

from kivy . uix . boxlayout import BoxLayout

from kivy . uix . button import Button

from kivy . uix . label import Label

from kivy . uix . textinput import TextInput

*class* LoginApp( *App* ):

*def* build( *self* ):

*self*.title = 'Se connecter'

        box = BoxLayout ( *orientation* ='horizontal ')

        box.add\_widget ( Label ( *text* ='Pin code '))

        box.add\_widget ( TextInput () )

        box.add\_widget ( Button ( *text* ='Entrer '))

        return box

# Configuration de la taille de la fenêtre

Config.set ('graphics ', 'width ', 350 )

Config.set ('graphics ', 'height ', 50 )

# Lancement de l’interface graphique

LoginApp().run()

=====================================================================================

box = BoxLayout ( *orientation* =’horizontal ’)

box . add\_widget ( Label ( *text* =’Pin code ’))

box . add\_widget ( TextInput () )

box . add\_widget ( Button ( *text* =’Entrer ’))

=====================================================================================

*class* GridGameApp ( *App* ):

*def* build ( *self* ):

        self . title = 'Grid Game '

        for i in range (12) :

            grid . add\_widget ( Button ( *text* = *str* (i + 1) ))

        return grid

=====================================================================================

from kivy.app import App

from kivy.uix.boxlayout import BoxLayout

from kivy.uix.button import Button

from kivy.uix.textinput import TextInput

*class* MainApp(*App*):

*def* build(*self*):

        self.operators = ["/", "\*", "+", "-"]

        self.last\_was\_operator = None

        self.last\_button = None

        main\_layout = BoxLayout(*orientation*="vertical")

        self.solution = TextInput(

*multiline*=False, *readonly*=True, *halign*="right", *font\_size*=55

        )

        main\_layout.add\_widget(self.solution)

        buttons = [

            ["7", "8", "9", "/"],

            ["4", "5", "6", "\*"],

            ["1", "2", "3", "-"],

            [".", "0", "C", "+"],

        ]

        for row in buttons:

            h\_layout = BoxLayout()

            for label in row:

                button = Button(

*text*=label,

*pos\_hint*={"center\_x": 0.5, "center\_y": 0.5},

                )

                button.bind(*on\_press*=self.on\_button\_press)

                h\_layout.add\_widget(button)

            main\_layout.add\_widget(h\_layout)

        equals\_button = Button(

*text*="=", *pos\_hint*={"center\_x": 0.5, "center\_y": 0.5}

        )

        equals\_button.bind(*on\_press*=self.on\_solution)

        main\_layout.add\_widget(equals\_button)

        return main\_layout

=====================================================================================

from kivy . app import App

from kivy . config import Config

from kivy . uix . boxlayout import BoxLayout

from kivy . uix . button import Button

from kivy . uix . label import Label

from kivy . uix . textinput import TextInput

*class* FreePosApp(*App*):

*def* build(*self*):

        self.title = 'Free Positioning '

        box = FloatLayout(*size*=(200, 150))

        box.add\_widget(Button(*text*='A', *size\_hint* = (0.3, 0.2), *pos*= (0, 0) ))

        box.add\_widget(Button(*text*='B', *size\_hint* = (0.3, 0.2), *pos* = (50, 80) ))

        return box

=====================================================================================

from kivymd.app import MDApp

from kivymd.uix.card import MDCard

from kivymd.uix.behaviors import RoundedRectangularElevationBehavior

from kivy.uix.screenmanager import ScreenManager,Screen

from kivymd.uix.fitimage import FitImage

from kivy.lang.builder import Builder

from kivymd.uix.widget import MDWidget

from kivymd.uix.toolbar import MDTopAppBar

from kivymd.uix.label import MDLabel

from kivy.metrics import dp

from kivy.uix.boxlayout import BoxLayout

from kivymd.uix.selectioncontrol import MDCheckbox

from kivymd.uix.textfield.textfield import MDTextField

from kivymd.theming import ThemableBehavior

from kivymd.uix.list import MDList,OneLineIconListItem

from kivymd.uix.list import IconLeftWidgetWithoutTouch

from kivy.uix.widget import Widget

from kivy.uix.button import Button

from kivy.core.window import Window

from multiprocessing import Process

from kivy.uix.scrollview import ScrollView

from kivymd.uix.boxlayout import MDBoxLayout

from kivy.uix.stacklayout import StackLayout

from kivymd.uix.navigationdrawer import MDNavigationDrawer,MDNavigationLayout

# Set the Window Size

Window.size=(1280,800)

# class My\_item\_elevation(MDCard,RoundedRectangularElevationBehavior):

#     pass

*class* Items\_of\_Menu(*MDBoxLayout*):

*def* \_\_init\_\_(*self*,\*\**kwargs*):

*super*(Items\_of\_Menu,self).\_\_init\_\_(\*\*kwargs)

        self.size\_hint=(1,0.9)

        scrollbarwin=ScrollView()

        content\_box=BoxLayout(*orientation*='vertical',*padding*=dp(8),*spacing*=dp(8),*size\_hint*=(1,None))

        content\_box.bind(*minimum\_height*=content\_box.setter('height'))

        # for loop which is used for adding the items in the as a list

        for i in range(0,100):

            Template\_card= Elevation(

*size\_hint\_y*=None,

*size\_hint\_x*=.960,

*height* = dp(100),

*padding* = dp(4),

*pos\_hint*={'center\_y': .5, 'center\_x': .490},

*radius* = [20,],

*elevation* = 4,

            )

            checkbox=MDCheckbox(

*size\_hint*=(None, None),

*size*= (dp(48),dp(48)),

*pos\_hint*={'center\_y': .5}

            )

            image\_box=MDBoxLayout(*adaptive\_size*=True)

            image=FitImage(

*source*="D:/Study/Python/Kivy/images/2.jpg",

*size\_hint*= (None, None),

*height*=dp(80),

*width*=dp(130),

*radius*=[12,],

*pos\_hint*={'center\_y':0.5}

            )

            image\_box.add\_widget(image)

            text\_box=MDBoxLayout(*orientation*="vertical",*adaptive\_height*=True,*pos\_hint*={'center\_y':0.5},*padding*=[12,0,0,0])

            item\_name=MDLabel(*text*="Hello,How are you? I think you are fine",*font\_style*="H5",*size\_hint*=(1,None),*bold*=True,*theme\_text\_color*="Primary")

            item\_name.bind(*texture\_size*=item\_name.setter('size'))

            price=MDLabel(*text*=*u*"Price: \u20B9400/per",*font\_style*="Subtitle1",*size\_hint*=(1,None),*bold*=True,*theme\_text\_color*="Hint")

            price.bind(*texture\_size*=price.setter('size'))

            quantitybox=MDBoxLayout(*orientation*='vertical',*adaptive\_height*=True,*size\_hint\_x*=0.2,*pos\_hint* = {'center\_y': .5,'center\_x':0.5})

            quantityfield=MDTextField(

*hint\_text*= "Quantity",

*mode*= "rectangle",

*size\_hint*=(None,None),

*width*=dp(80),

*height*= dp(40),

*padding*=[0,0,15,0]

            )

            quantitybox.add\_widget(quantityfield)

            Template\_card.add\_widget(checkbox)

            Template\_card.add\_widget(image\_box)

            Template\_card.add\_widget(text\_box)

            text\_box.add\_widget(item\_name)

            text\_box.add\_widget(price)

            Template\_card.add\_widget(quantitybox)

            content\_box.add\_widget(Template\_card)

        scrollbarwin.add\_widget(content\_box)

        self.add\_widget(scrollbarwin)

*class* navigation(*Screen*):

*def* \_\_init\_\_(*self*,\*\**kwargs*):

*super*(navigation,self).\_\_init\_\_(\*\*kwargs)

        # Navigation Layout

        nav\_layout=MDNavigationLayout()

        # Theme Color

        theme=ThemableBehavior()

        # Screen Manager

        scr\_mang=ScreenManager()

        # Navigation Drawer

        nav\_drawer=MDNavigationDrawer()

        # List of Items

        self.list\_for\_item=MDList()

        # Put Items in List

        #1

        icon\_image = IconLeftWidgetWithoutTouch(*icon*="food")

        item\_list = OneLineIconListItem(*text*="Main Course", *theme\_text\_color*="Custom", *text\_color*=theme.theme\_cls.text\_color)

        item\_list.add\_widget(icon\_image)

        item\_list.bind(*on\_release*=*lambda* *x*: self.Change\_Screen(x))

        self.list\_for\_item.add\_widget(item\_list)

        #2

        icon\_image2 = IconLeftWidgetWithoutTouch(*icon*="coffee")

        item2\_list = OneLineIconListItem(*text*="Tea and Coffee", *theme\_text\_color*="Custom", *text\_color*=theme.theme\_cls.text\_color)

        item2\_list.add\_widget(icon\_image2)

        item2\_list.bind(*on\_release*=*lambda* *x*: self.Change\_Screen(x))

        self.list\_for\_item.add\_widget(item2\_list)

        #3

        icon\_image3 = IconLeftWidgetWithoutTouch(*icon*="ice-cream")

        item3\_list = OneLineIconListItem(*text*="Ice Creams", *theme\_text\_color*="Custom", *text\_color*=theme.theme\_cls.text\_color)

        item3\_list.add\_widget(icon\_image3)

        item3\_list.bind(*on\_release*=*lambda* *x*: self.Change\_Screen(x))

        self.list\_for\_item.add\_widget(item3\_list)

        #4

        icon\_image4 = IconLeftWidgetWithoutTouch(*icon*="bottle-soda")

        item4\_list = OneLineIconListItem(*text*="Cold Drinks", *theme\_text\_color*="Custom", *text\_color*=theme.theme\_cls.text\_color)

        item4\_list.add\_widget(icon\_image4)

        item4\_list.bind(*on\_release*=*lambda* *x*: self.Change\_Screen(x))

        self.list\_for\_item.add\_widget(item4\_list)

        #5

        icon\_image5 = IconLeftWidgetWithoutTouch(*icon*="hamburger")

        item5\_list = OneLineIconListItem(*text*="Fast Food", *theme\_text\_color*="Custom", *text\_color*=theme.theme\_cls.text\_color)

        item5\_list.add\_widget(icon\_image5)

        item5\_list.bind(*on\_release*=*lambda* *x*: self.Change\_Screen(x))

        self.list\_for\_item.add\_widget(item5\_list)

        #6

        icon\_image6 = IconLeftWidgetWithoutTouch(*icon*="food-takeout-box")

        item6\_list = OneLineIconListItem(*text*="Special Dishes", *theme\_text\_color*="Custom", *text\_color*=theme.theme\_cls.text\_color)

        item6\_list.add\_widget(icon\_image6)

        item6\_list.bind(*on\_release*=*lambda* *x*: self.Change\_Screen(x))

        self.list\_for\_item.add\_widget(item6\_list)

        # Whole Content box is here

        content\_box=MDBoxLayout(*orientation*="vertical",*spacing*=dp(8),*padding*=dp(8))

        # Screen Stuff is here

        items\_part=Items\_of\_Menu()

        # Hotel Information

        hotel\_image=FitImage(*source*="Hotel logo.jpg",*radius*=[100,100,100,100],*size\_hint*=(None,None),*size*=(dp(100),dp(100)),*pos\_hint*={"center\_x":0.5})

        hotel\_name=MDLabel(*text*="Hotel Heera Panna and Family Restaurant",*size\_hint*=(1,None))

        hotel\_name.font\_style="Subtitle2"

        hotel\_name.bind(*texture\_size*=hotel\_name.setter("size"))

        # Email Info

        email = MDLabel(*text*="17288hdkdndk@gmail.com", *size\_hint*=(1,None))

        email.bind(*texture\_size*=email.setter("size"))

        email.font\_style="Caption"

        # ScrollBar for Items of List in Navigation Drawer

        nav\_scroll=ScrollView()

        # Screen which is a child of  Whole Content Box

        scr=Screen()

        # the Top Toolbar  of the Window(Header)

        box=MDBoxLayout(*orientation*="vertical")

        self.toolbar=MDTopAppBar(*title*="Demo App",*left\_action\_items*=[["menu",*lambda* *x*:nav\_drawer.set\_state("open")]],*elevation*=10)

        # Adding the Child Widget to the Parent Widget

        box.add\_widget(self.toolbar)

        box.add\_widget(items\_part)

        scr.add\_widget(box)

        scr\_mang.add\_widget(scr)

        nav\_layout.add\_widget(scr\_mang)

        content\_box.add\_widget(hotel\_image)

        content\_box.add\_widget(hotel\_name)

        content\_box.add\_widget(email)

        nav\_scroll.add\_widget(self.list\_for\_item)

        content\_box.add\_widget(nav\_scroll)

        nav\_drawer.add\_widget(content\_box)

        nav\_layout.add\_widget(nav\_drawer)

        self.add\_widget(nav\_layout)

*def* Change\_Screen(*self*,*instance*):

        value=instance.text

        if MymdCard.sm.current\_screen!=value:

            MymdCard.sm.current=value

*class* mainHeading(*MDWidget*,*Widget*):

    pass

*class* template:

    pass

*class* Elevation(*RoundedRectangularElevationBehavior*,*MDCard*):

    pass

*class* FirstWin(*RoundedRectangularElevationBehavior*,*Screen*,*mainHeading*):

    textAndImageReference={"main\_txt":[],"image":[],"hint\_txt":[]}

    Category\_names=["Main Course","Tea and Coffee","Ice Creams","Cold Drinks","Fast Food","Special Dishes"]

*def* \_\_init\_\_(*self*,\*\**kwargs*):

*super*(FirstWin,self).\_\_init\_\_(\*\*kwargs)

        scrollbar=ScrollView(*size\_hint\_y*=None,*pos\_hint*={'x':0,'top':0.850},*size*=(Window.width,Window.height))

        secondary\_widget=StackLayout(*size\_hint*=(1,None),*spacing*=50,*padding*=20)

        secondary\_widget.bind(*minimum\_height*=secondary\_widget.setter('height'))

        for i in range(0,len(self.Category\_names)):

            mycard=Elevation(

*elevation*=15,

*size\_hint* =(0.2,None),

*height*=350,

*orientation*='vertical',

*radius*= [36, ],

*ripple\_behavior*=True,

*focus\_behavior*=True

            )

            mycard.bind(*size*=self.adjust\_sizes)

            image = FitImage(*radius*=[36,36,0,0],*size\_hint\_y*=3, *size\_hint\_x*=1,*orientation*="vertical")

            imagebutton = Button(*background\_normal*="D:/Study/Python/Kivy/images/1.jpg",

*background\_down*="D:/Study/Python/Kivy/images/1.jpg",

*size\_hint\_y*=550.0,

*size\_hint\_x*=1,

*pos\_hint*={'x': 0, 'y': 0}

                                 )

            self.textAndImageReference["image"].append(imagebutton)

            imagebutton.bind(*on\_release*=*lambda* *x*:self.Change\_Menu\_category(x))

            texture\_part = MDBoxLayout( *md\_bg\_color*=(46 / 255, 8 / 255, 211 / 255, .5),

*radius*=[0, 0, 36, 36],*orientation*="vertical")

            main\_text = Button(

*text*=self.Category\_names[i],

*halign*="center",

*bold*=True,

*font\_size*=mycard.width /6,

*background\_normal*='',

*background\_color*=(0, 0, 0, 0)

            )

            self.textAndImageReference["main\_txt"].append(main\_text)

            main\_text.bind(*on\_release*=*lambda* *x*:self.Change\_Menu\_category(x))

            Hint\_text = Button(

*text*="Food Menu",

*halign*="center",

*font\_size*=mycard.width/6,

*bold*=True,

*color*=(206 / 255, 203 / 255, 203 / 255, 0.2),

*background\_normal*='',

*background\_color*=(0, 0, 0, 0)

            )

            self.textAndImageReference["hint\_txt"].append(Hint\_text)

            Hint\_text.bind(*on\_release*=*lambda* *x*:self.Change\_Menu\_category(x))

            image.add\_widget(imagebutton)

            mycard.add\_widget(image)

            texture\_part.add\_widget(main\_text)

            texture\_part.add\_widget(Hint\_text)

            mycard.add\_widget(texture\_part)

            secondary\_widget.add\_widget(mycard)

        last\_one=MDBoxLayout(*size\_hint*=(1,None),*height*=20)

        secondary\_widget.add\_widget(last\_one)

        scrollbar.add\_widget(secondary\_widget)

        self.add\_widget(scrollbar)

*def* adjust\_sizes(*self*, *mycard*, *new\_size*):

        for i in range(0,len(self.textAndImageReference["main\_txt"])):

            self.textAndImageReference["main\_txt"][i].font\_size =mycard.width/10

            self.textAndImageReference["hint\_txt"][i].font\_size =mycard.width/15

*def* Change\_Menu\_category(*self*,*instance*):

        for i in range(0,len(self.textAndImageReference["main\_txt"])):

            if instance==self.textAndImageReference["main\_txt"][i]:

                item=self.textAndImageReference["main\_txt"][i].text

                MymdCard.sm.current=item

            elif instance==self.textAndImageReference["hint\_txt"][i]:

                item=self.textAndImageReference["main\_txt"][i].text

                MymdCard.sm.current=item

            elif instance==self.textAndImageReference["image"][i]:

                item=self.textAndImageReference["main\_txt"][i].text

                MymdCard.sm.current=item

*class* Main\_Course(*navigation*):

*def* \_\_init\_\_(*self*,\*\**kwargs*):

*super*(Main\_Course,self).\_\_init\_\_(\*\*kwargs)

        name="Main Course"

        color=ThemableBehavior()

        self.toolbar.title=name

        for item in self.list\_for\_item.children:

            if item.text==name:

                item.text\_color=color.theme\_cls.primary\_color

*class* Tea\_Coffee(*navigation*):

*def* \_\_init\_\_(*self*, \*\**kwargs*):

*super*(Tea\_Coffee, self).\_\_init\_\_(\*\*kwargs)

        name = "Tea and Coffee"

        color = ThemableBehavior()

        self.toolbar.title = name

        for item in self.list\_for\_item.children:

            if item.text == name:

                item.text\_color = color.theme\_cls.primary\_color

*class* Ice\_creams(*navigation*):

*def* \_\_init\_\_(*self*, \*\**kwargs*):

*super*(Ice\_creams, self).\_\_init\_\_(\*\*kwargs)

        name = "Ice Creams"

        color = ThemableBehavior()

        self.toolbar.title = name

        for item in self.list\_for\_item.children:

            if item.text == name:

                item.text\_color = color.theme\_cls.primary\_color

*class* Cold\_drinks(*navigation*):

*def* \_\_init\_\_(*self*, \*\**kwargs*):

*super*(Cold\_drinks, self).\_\_init\_\_(\*\*kwargs)

        name = "Cold Drinks"

        color = ThemableBehavior()

        self.toolbar.title = name

        for item in self.list\_for\_item.children:

            if item.text == name:

                item.text\_color = color.theme\_cls.primary\_color

*class* Fast\_food(*navigation*):

*def* \_\_init\_\_(*self*, \*\**kwargs*):

*super*(Fast\_food, self).\_\_init\_\_(\*\*kwargs)

        name = "Fast Food"

        color = ThemableBehavior()

        self.toolbar.title = name

        for item in self.list\_for\_item.children:

            if item.text == name:

                item.text\_color = color.theme\_cls.primary\_color

*class* Special\_dishes(*navigation*):

*def* \_\_init\_\_(*self*, \*\**kwargs*):

*super*(Special\_dishes, self).\_\_init\_\_(\*\*kwargs)

        name = "Special Dishes"

        color = ThemableBehavior()

        self.toolbar.title = name

        for item in self.list\_for\_item.children:

            if item.text == name:

                item.text\_color = color.theme\_cls.primary\_color

*class* MymdCard(*MDApp*):

    sm=ScreenManager()

*def* build(*self*):

        Builder.load\_file("md\_card\_py.kv")

        self.theme\_cls.theme\_style = "Dark"

        Screens=[FirstWin(*name*='Menu\_category'),Main\_Course(*name*="Main Course"),Tea\_Coffee(*name*="Tea and Coffee"),Ice\_creams(*name*="Ice Creams"),Cold\_drinks(*name*="Cold Drinks"),Fast\_food(*name*="Fast Food"),Special\_dishes(*name*="Special Dishes")]

        for screen in Screens:

            self.sm.add\_widget(screen)

        return self.sm

if \_\_name\_\_ == '\_\_main\_\_':

    MymdCard().run()

=====================================================================================

from kivy.app import App

from kivy.uix.boxlayout import BoxLayout

from kivy.uix.button import Button

from kivy.uix.textinput import TextInput

*class* MainApp(*App*):

*def* build(*self*):

        self.operators = ["/", "\*", "+", "-"]

        self.last\_was\_operator = None

        self.last\_button = None

        main\_layout = BoxLayout(*orientation*="vertical")

        self.solution = TextInput(

*multiline*=False, *readonly*=True, *halign*="right", *font\_size*=55

        )

        main\_layout.add\_widget(self.solution)

        buttons = [

            ["7", "8", "9", "/"],

            ["4", "5", "6", "\*"],

            ["1", "2", "3", "-"],

            [".", "0", "C", "+"],

        ]

        for row in buttons:

            h\_layout = BoxLayout()

            for label in row:

                button = Button(

*text*=label,

*pos\_hint*={"center\_x": 0.5, "center\_y": 0.5},

                )

                button.bind(*on\_press*=self.on\_button\_press)

                h\_layout.add\_widget(button)

            main\_layout.add\_widget(h\_layout)

        equals\_button = Button(

*text*="=", *pos\_hint*={"center\_x": 0.5, "center\_y": 0.5}

        )

        equals\_button.bind(*on\_press*=self.on\_solution)

        main\_layout.add\_widget(equals\_button)

        return main\_layout

=====================================================================================

import kivy

from kivy.app import App

from kivy.uix.label import Label

from kivy.uix.gridlayout import GridLayout

from kivy.uix.textinput import TextInput

*class* MyGrid(*GridLayout*):

*def* \_\_init\_\_(*self*, \*\**kwargs*):

*super*(MyGrid, self).\_\_init\_\_(\*\*kwargs)

        self.cols = 2

        self.add\_widget(Label(*text*="First Name: "))

        self.name = TextInput(*multiline*=False)

        self.add\_widget(self.name)

        self.add\_widget(Label(*text*="Last Name: "))

        self.lastName = TextInput(*multiline*=False)

        self.add\_widget(self.lastName)

        self.add\_widget(Label(*text*="Email: "))

        self.email = TextInput(*multiline*=False)

        self.add\_widget(self.email)

*class* MyApp(*App*):

*def* build(*self*):

        return MyGrid()

MyApp().run()

=====================================================================================

WINDOW :

from kivy.app import App

from kivy.uix.label import Label

from kivy.core.window import Window

Window.clearcolor = (100/255.0,0,2,1)

Window.size=(370,600)

*class* App1(*App*):

    pass

if \_\_name\_\_== '\_\_main\_\_':

    App1().run()

Window.clearcolor = (89/255.0,7/255.0,45/255.0,177/255.0)

LABEL:

*class* Myapp(*App*):

*def* build(*self*):

        return Label( *text* = "ok python",

*color* = (.56,0,1,1),

*font\_size*=30

            )

Ou :

*class* Name(*App*):

    pass

if \_\_name\_\_== '\_\_main\_\_':

    Name().run()

*class* Name(*App*):

*def* build(*self*):

        pass

if \_\_name\_\_== '\_\_main\_\_':

    Name().run()

BUTTON :

from kivy.app import App

from kivy.core.window import Window

from kivy.uix.button import Button

Window.clearcolor = (89/255.0,7/255.0,45,3)

Window.size=(370,600)

*class* Myapp(*App*):

*def* build(*self*):

        b1=Button(

*text*="Home",

*size\_hint*=(0.3,0.1),

*font\_size*='22',

*pos\_hint*={'x':0.35,'y':0.19},

*color*=(1,.21,0,1),

*background\_color*=(0,.60,.3,1)

        )

        return b1

if \_\_name\_\_== '\_\_main\_\_':

    Myapp().run()

Ou :

Button:

    text:'Home'

    size\_hint:(0.3,0.1)

    font\_size:'22'

    pos\_hint:{'x':0.35,'y':0.19}

    color:(1,1,1,1)

    background\_color:(.40,.10,.30,1)

=======================================================================

from kivy.app import App

from kivy.core.window import Window

from kivy.uix.button import Button

Window.clearcolor = (100/255.0,0,0,3)

Window.size=(370,600)

*class* Name(*App*):

    pass

if \_\_name\_\_== '\_\_main\_\_':

    Name().run()

*class* Myapp(*App*):

*def* build(*self*):

        b1=Button(

*text*="Home",

*size\_hint*=(0.3,0.1),

*font\_size*='22',

*pos\_hint*={'x':0.35,'y':0.19},

*color*=(1,.21,0,1),

*background\_color*=(0,.60,.3,1),

*on\_press* = self.clickyes,

*on\_release* = self.clickno

        )

        return b1

*def* clickyes(*self*,*yes*):

        print("yes")

*def* clickno (*self*,*no*):

        print('no')

====================================================================

TextInput :

*class* Myapp(*App*):

*def* build(*self*):

        texo = TextInput(

*text* = 'Enter your email: \n ',

*multiline*= False,

*font\_size*=16,

*pos\_hint*={'x':0.2,'y':0.6},

*size\_hint*=(0.6,0.2)

        )

        return texo

========================================================

Image:

*class* Myapp(*App*):

*def* build(*self*):

        imj=Image(

*source* ='img.png'

        )

        return imj

=======================================

*class* Myapp(*App*):

*def* build(*self*):

        imj=Image(

*source* ='img.png',

*size\_hint* = (0.2,0.2),

*pos\_hint*={'x':0.4,'y':0.4}

        )

        return imj

Image en lien :

        from kivy.uix.image import AsyncImage

        return AsyncImage(*source*='http://kivy.org/logos/kivy-logo-black-64.png')

=======================================

Checkbox:

*class* Myapp(*App*):

*def* build(*self*):

        chek=CheckBox()

        chek.bind(*active*=clicko)

        return chek

*def* clicko(*chek*,*ok*):

        print('ok')

if \_\_name\_\_== '\_\_main\_\_':

    Myapp().run()

=======================================

*class* Myapp(*App*):

*def* build(*self*):

        chek=CheckBox()

        chek.bind(*active*=clicko)

        return chek

*def* clicko(*chek*,*ok*):

    if ok :

        print('ok')

    else:

        print('not ok')

if \_\_name\_\_== '\_\_main\_\_':

    Myapp().run()

==============================================================================

ACTIONBAR:

*class* Myapp(*App*):

*def* build(*self*):

        pass

runTouchApp(Builder.load\_string('''

ActionBar:

    pos\_hint:{'top':1}

    ActionView:

        ActionPrevious:

            title:'MY FIRST APP'

        ActionButton:

            text:'Home'

        ActionButton:

            text:'Back'

        ActionGroup:

            text:'More'

            color:.3,.6,2,1

            ActionButton:

                text:'bt1'

            ActionButton:

                text:'bt2'

'''))

 ==============================================================================

*class* Myapp(*App*):

*def* build(*self*):

        img=Image(*source*='img3.png')

        return img

runTouchApp(Builder.load\_string('''

ActionBar:

    pos\_hint:{'top':1}

    ActionView:

        ActionPrevious:

            title:'Universite Ibn Tofail'

            color:.4,.6,2,1

        ActionButton:

            text:'Home'

        ActionButton:

            text:'Back'

            color:100/255.0,0,0,3

            background\_color: 1,1,1,3

        ActionGroup:

            text:'More'

            color:1,1,2,1

            ActionButton:

                text:'EST'

                color:.3,.7,2,1

            ActionButton:

                text:'FS'

                color:.7,.8,2,2

            ActionButton:

                text:'ENSA'

                color:.2,.9,2,4

            ActionButton:

                text:'ENCG'

                color:.1,.6,2,3

'''))

=====================================================================================

import kivy

from kivy.app import App

from kivy.uix.textinput import TextInput

from kivy.uix.label import Label

from kivy.uix.image import Image

from kivy.uix.button import Button

from kivy.uix.gridlayout import GridLayout

from kivy.core.audio import SoundLoader

from kivy.lang import Builder

from kivy.uix.screenmanager import Screen

from kivy.uix.screenmanager import ScreenManager

from kivy.base import runTouchApp

from glob import glob

from random import randint

from os.path import join , dirname

from kivy.uix.scatter import Scatter

from kivy.properties import StringProperty

from kivy.core.window import Window

=====================================================================================

import kivy

from kivy.app import App

from kivy.uix.textinput import TextInput

from kivy.uix.label import Label

from kivy.uix.image import Image

from kivy.uix.button import Button

from kivy.uix.gridlayout import GridLayout

from kivy.core.audio import SoundLoader

from kivy.lang import Builder

from kivy.uix.screenmanager import Screen

from kivy.uix.screenmanager import ScreenManager

from kivy.base import runTouchApp

from glob import glob

from random import randint

from os.path import join , dirname

from kivy.uix.scatter import Scatter

from kivy.properties import StringProperty

from kivy.core.window import Window

from kivy.uix.boxlayout import BoxLayout

import sqlite3

Window.clearcolor = (89/255.0,7/255.0,45,3)

Window.size=(400,600)

*class* Demo(*App*):

*def* build(*self*):

        conn = sqlite3.connect('frst.db')

        c=conn.cursor()

        c.execute("""CREATE TABLE if not exists cust (name text) """)

        conn.commit()

        conn.close()

        return Builder.load\_file('frst.kv')

*def* submit(*self*):

        conn = sqlite3.connect('frst.db')

        c=conn.cursor()

        c.execute("INSERT INTO cust VALUES (:frst)",

          {

          'frst': self.root.ids.word\_input.text,

          } )

        self.root.ids.word\_label.text = *f*'{self.root.ids.word\_input.text } Added'

        self.root.ids.word\_input.text=''

        conn.commit()

        conn.close()

*def* ok(*self*):

        conn = sqlite3.connect('frst.db')

        c=conn.cursor()

        c.execute("SELECT \* FROM cust")

        records=c.fetchall()

        word=''

        for record in records:

            word= *f*' { word}\n{record[0]}'

            self.root.ids.word\_label.text = *f*' {word}'

        conn.commit()

        conn.close()

if \_\_name\_\_== '\_\_main\_\_':

    Demo().run()

=====================================================================================

FloatLayout:

    BoxLayout:

        orientation: 'vertical'

        size: root.width, root.height

        Label:

            id: word\_label

            text\_size: self.size

            text: "           Enter A Name:\n\n\n"

            font\_hint:(0.1,0.9)

            font\_size: 32

        TextInput:

            id: word\_input

            multiline: False

            size\_hint:(1, .5)

        Button:

            size\_hint:(1, .5)

            font\_size:32

            text: "submit"

            on\_press: app.submit()

        Button:

            size\_hint:(1, .5)

            font\_size:32

            text: "ok2"

            on\_press: app.ok()

=====================================================================================

Wm:

    W1:

    W2:

    Est:

    Ensa:

    Encg:

    Fs:

    Not:

    Etu:

    Aut:

    Error:

    Mod:

    Res:

<W1>:

    name:'W1'

    GridLayout:

        cols:1

        GridLayout:

            rows:4

            Image:

                source:'img3.png'

            Label:

                text: 'LOGING'

                color: 1 , 0 , 0 , 1

                font\_size: 22

            Label:

                text: 'PLEASE ENTER PASSWORD:'

                font\_hint:(0.1,0.3)

            TextInput:

                id:passw

                text:''

                multiline: False

                size\_hint:(0.1,0.3)

        Button:

            text:'LOGIN'

            size\_hint:(0.1,0.1)

            on\_release:

                app.root.current='Mod' if passw.text=='1' else 'Error'

                root.manager.transition.direction='up'

<W2>:

    name:'W2'

    GridLayout:

        cols:1

        GridLayout:

            rows:2

            Image:

                source:'img3.png'

                size:(30,30)

            Label:

                text:'WELCOME'

        GridLayout:

            cols:2

            rows:3

            Button:

                text: 'EST'

                on\_release:

                    app.root.current='Est'

                    root.manager.transition.direction='up'

            Button:

                text: 'ENSA'

                on\_release:

                    app.root.current='Ensa'

                    root.manager.transition.direction='up'

            Button:

                text: 'FS'

                on\_release:

                    app.root.current='Fs'

                    root.manager.transition.direction='up'

            Button:

                text: 'ENCG'

                on\_release:

                    app.root.current='Encg'

                    root.manager.transition.direction='up'

        Button:

            text:'GO BACK'

            size\_hint:.1,0.3

            on\_release:

                app.root.current='W1'

                root.manager.transition.direction='down'

=================================================================================

<Est>:

    name:'Est'

    GridLayout:

        cols:1

        GridLayout:

            rows:4

            Image:

                source:'img1.png'

            Label:

                text: 'WELCOME TO EST KENITRA'

                color: 1 , 1 , 1 , 1

                font\_size: 22

        GridLayout:

            cols:2

            rows:3

            Button:

                text: 'LES ETUDIENTS'

                on\_release:

                    app.root.current='Est'

                    root.manager.transition.direction='up'

            Button:

                text: 'LES NOTES'

                on\_release:

                    app.root.current='Ensa'

                    root.manager.transition.direction='up'

            Button:

                text: 'LES MODULES'

                on\_release:

                    app.root.current='Mod'

                    root.manager.transition.direction='up'

            Button:

                text: 'VALIDATION'

                on\_release:

                    app.root.current='Encg'

                    root.manager.transition.direction='up'

        Button:

            text:'GO BACK'

            size\_hint:.1,0.3

            on\_release:

                app.root.current='W2'

                root.manager.transition.direction='down'

=================================================================================

<Fs>:

    name:'Fs'

    GridLayout:

        cols:1

        GridLayout:

            rows:4

            Image:

                source:'img1.png'

            Label:

                text: 'WELCOME TO LA FAC DE SCIENCE, KENITRA'

                color: 1 , 1 , 1 , 1

                font\_size: 22

        GridLayout:

            cols:2

            rows:3

            Button:

                text: 'LES ETUDIENTS'

                on\_release:

                    app.root.current='Est'

                    root.manager.transition.direction='up'

            Button:

                text: 'LES NOTES'

                on\_release:

                    app.root.current='Ensa'

                    root.manager.transition.direction='up'

            Button:

                text: 'LES MODULES'

                on\_release:

                    app.root.current='Fs'

                    root.manager.transition.direction='up'

            Button:

                text: 'VALIDATION'

                on\_release:

                    app.root.current='Encg'

                    root.manager.transition.direction='up'

        Button:

            text:'GO BACK'

            size\_hint:.1,0.3

            on\_release:

                app.root.current='W2'

                root.manager.transition.direction='down'

=================================================================================

<Encg>:

    name:'Encg'

    GridLayout:

        cols:1

        GridLayout:

            rows:4

            Image:

                source:'img1.png'

            Label:

                text: 'WELCOME TO EST KENITRA'

                color: 1 , 1 , 1 , 1

                font\_size: 22

        GridLayout:

            cols:2

            rows:3

            Button:

                text: 'LES ETUDIENTS'

                on\_release:

                    app.root.current='Est'

                    root.manager.transition.direction='up'

            Button:

                text: 'LES NOTES'

                on\_release:

                    app.root.current='Ensa'

                    root.manager.transition.direction='up'

            Button:

                text: 'LES MODULES'

                on\_release:

                    app.root.current='Mod'

                    root.manager.transition.direction='up'

            Button:

                text: 'VALIDATION'

                on\_release:

                    app.root.current='Encg'

                    root.manager.transition.direction='up'

        Button:

            text:'GO BACK'

            size\_hint:.1,0.3

            on\_release:

                app.root.current='W2'

                root.manager.transition.direction='down'

#=================================================================================

<Ensa>:

    name:'Ensa'

    GridLayout:

        cols:1

        GridLayout:

            rows:4

            Image:

                source:'img1.png'

            Label:

                text: 'WELCOME TO EST KENITRA'

                color: 1 , 1 , 1 , 1

                font\_size: 22

        GridLayout:

            cols:2

            rows:3

            Button:

                text: 'LES ETUDIENTS'

                on\_release:

                    app.root.current='Est'

                    root.manager.transition.direction='up'

            Button:

                text: 'LES NOTES'

                on\_release:

                    app.root.current='Ensa'

                    root.manager.transition.direction='up'

            Button:

                text: 'LES MODULES'

                on\_release:

                    app.root.current='Fs'

                    root.manager.transition.direction='up'

            Button:

                text: 'VALIDATION'

                on\_release:

                    app.root.current='Encg'

                    root.manager.transition.direction='up'

        Button:

            text:'GO BACK'

            size\_hint:.1,0.3

            on\_release:

                app.root.current='W2'

                root.manager.transition.direction='down'

#=================================================================================

<Not>:

    name:'Not'

    GridLayout:

        cols:1

        GridLayout:

            rows:4

            Image:

                source:'img21.jpg'

            Label:

                text: 'LES NOTES'

                color: 1 , 1 , 1 , 1

                font\_size: 22

#=================================================================================

<Etu>:

    name:'Etu'

    GridLayout:

        cols:1

        GridLayout:

            rows:4

            Image:

                source:'img21.jpg'

            Label:

                text: 'LES ETUDIENTS'

                color: 1 , 1 , 1 , 1

                font\_size: 22

#=================================================================================

#=================================================================================

<Mod>:

    name:'Mod'

    GridLayout:

        rows:2

        Label:

            id:LMOD

            text:'ENTREZ LES NOTES DES MODULES DE S1: '

            color: .89 , .97 , .130 , 1

            font\_size: 16

        GridLayout:

            cols:2

            rows:5

            Label:

                text:'MODELE: 1'

            TextInput:

                id:M1

                text:'hn'

            Label:

                text:'MODELE: 2'

            TextInput:

                id:M2

                text:''

            Label:

                text:'MODELE: 3'

            TextInput:

                id:M3

                text:''

            Label:

                text:'MODELE: 4'

            TextInput:

                id:M4

                text:''

            Button:

                text:'RESULTAT:'

                on\_release:

                    app.root.current='Res' if *int*(M1.text)>=12 else 'Mod'

                    root.manager.transition.direction='up'

<Res>:

    name:'Res'

    GridLayout:

        rows:2

        Label:

            text:'VALIDEE ,,'

            color: .89 , .97 , .130 , 1

            font\_size: 16

        Button:

            text:'BACK'

            on\_release:

                app.root.current= 'Mod'

                root.manager.transition.direction='down'

#=================================================================================

<Aut>:

    name:'Aut'

    GridLayout:

        cols:1

        GridLayout:

            rows:4

            Image:

                source:'img21.jpg'

            Label:

                text: 'AUTRE'

                color: 1 , 1 , 1 , 1

                font\_size: 22

#=================================================================================

<Error>:

    name:'Error'

    GridLayout:

        cols:1

        GridLayout:

            rows:4

            Image:

                source:'img2.png'

            Label:

                text: 'ERROR IN PASSWORD !!! '

                color: 100/255.0 , 4 , 0.6 , 1

                font\_size: 22

            Button:

                text:' TRY AGAINE !!'

                size\_hint:0.1,0.3

                on\_release:

                    app.root.current='W1'

                    root.manager.transition.direction='down'

=================================================================================

from kivy.app import App

from kivy.lang import Builder

from kivy.uix.boxlayout import BoxLayout

mykv = Builder.load\_string("""

<MyLabels>:

Label:

text: root.my\_any()

Label:

text: '2'

""")

class MyLabels(BoxLayout):

def my\_any(self):

print('in my\_any')

w = 'this is a string'

return w

class ExampleApp(App):

def build(self):

return MyLabels()

if \_\_name\_\_ == '\_\_main\_\_':

ExampleApp().run()

Database1:

#=================================================================================

from kivy.lang import Builder  
from kivy.core.window import Window  
from kivy.lang import Builder  
from kivymd.app import MDApp  
import mysql.connector  
  
Window.clearcolor = (89/255.0,7/255.0,45,3)  
Window.size=(400,600)  
  
class MainApp(MDApp):  
  
 def build(self):  
 self.theme\_cls.theme\_style="Dark"  
 self.theme\_cls.primary\_palette = "BlueGray"  
 return Builder.load\_file('second\_db.kv')  
  
 def submit(self):  
 mydb = mysql.connector.connect(  
 host="localhost",  
 user="root",  
 passwd="11121314",  
 database="second\_db"  
  
 )  
 c = mydb.cursor()  
 cmd="INSERT INTO etudiant (name) VALUES (%s)"  
 val= (self.root.ids.word\_input.text,)  
 c.execute(cmd, val)  
 self.root.ids.word\_label.text=f'{self.root.ids.word\_input.text} ADDED '  
 self.root.ids.word\_input.text=''  
  
 mydb.commit()  
 mydb.close()  
  
 def show (self):  
 mydb = mysql.connector.connect(  
 host="localhost",  
 user="root",  
 passwd="11121314",  
 database="second\_db"  
  
 )  
 c = mydb.cursor()  
 c.execute("select name from etudiant")  
 rec=c.fetchall()  
 word=''  
  
 for rec in rec:  
 word = f'{word}\n{rec[0]}'  
 self.root.ids.word\_label.text=f'{word}'  
  
 mydb.commit()  
 mydb.close()  
  
  
if \_\_name\_\_== '\_\_main\_\_':  
 MainApp().run()

kivy file:

GridLayout:  
 cols:1   
 GridLayout:  
 rows:4  
 Label:  
 id: word\_label  
 text\_size: self.size  
 text: " Enter A Name:\n\n\n"  
 font\_hint:(0.1,0.3)   
 font\_size: 32  
  
   
 TextInput:  
 id: word\_input  
 multiline: False  
 size\_hint:(1, .5)  
  
   
   
 Button:  
 size\_hint:(1, .5)  
 font\_size:25  
 text: "submit"  
 on\_press: app.submit()  
   
  
 Button:  
 size\_hint:(1, .5)  
 font\_size:24  
 text: "ok2"  
 on\_press: app.show()

#=================================================================================

from kivy.lang import Builder  
from kivy.core.window import Window  
from kivy.lang import Builder  
from kivymd.app import MDApp  
import mysql.connector  
  
Window.clearcolor = (89/255.0,7/255.0,45,3)  
Window.size=(400,600)  
  
class MainApp(MDApp):  
  
 def build(self):  
 self.theme\_cls.theme\_style="Dark"  
 self.theme\_cls.primary\_palette = "BlueGray"  
  
  
 return Builder.load\_file('second\_db.kv')  
  
 def submit(self):  
 mydb = mysql.connector.connect(  
 host="localhost",  
 user="root",  
 passwd="11121314",  
 database="second\_db")  
  
  
  
 c = mydb.cursor()  
 cmd="INSERT INTO etudiant (name) VALUES (%s)"  
 val= (self.root.ids.word\_input.text,)  
 c.execute(cmd, val)  
 self.root.ids.word\_label.text=f'{self.root.ids.word\_input.text} ADDED '  
 self.root.ids.word\_input.text=''  
  
 mydb.commit()  
 mydb.close()  
  
 def show (self):  
 mydb = mysql.connector.connect(  
 host="localhost",  
 user="root",  
 passwd="11121314",  
 database="second\_db")  
  
  
 c = mydb.cursor()  
 c.execute("select name from etudiant")  
 rec=c.fetchall()  
 word=''  
  
 for rec in rec:  
 word = f'{word}\n{rec[0]}'  
 self.root.ids.word\_label.text=f'{word}'  
  
 mydb.commit()  
 mydb.close()  
  
  
if \_\_name\_\_== '\_\_main\_\_':  
 MainApp().run()

#=================================================================================

#=================================================================================

#=================================================================================

#=================================================================================

#=================================================================================

=====================================================================================

#=================================================================================

#=================================================================================

#=================================================================================

#=================================================================================

=====================================================================================

#=================================================================================

#=================================================================================

#=================================================================================

#=================================================================================

=====================================================================================