

Kivy #3 KV Language

1. ติดตั้ง Kivy ตามใบงานก่อนหน้าให้เรียบร้อย
2. สร้างโค้ดด้านล่าง แล้วทดลองรัน

```
from kivy.app import App
from kivy.core.window import Window

from kivy.uix.scatter import Scatter
from kivy.uix.label import Label
from kivy.uix.floatlayout import FloatLayout
from kivy.uix.textinput import TextInput
from kivy.uix.boxlayout import BoxLayout

class TutorialApp(App):
    def build(self):
        b = BoxLayout(orientation='vertical')
        t = TextInput(text='Hello',
                      font_size=150,
                      size_hint_y=None,
                      height=200)
        l = Label(font_size=150,
                  text='Hello')
        f = FloatLayout()
        size = Window.size
        s = Scatter(pos=(90, size[1]/2 - 200))

        f.add_widget(s)
        s.add_widget(l)

        b.add_widget(t)
        b.add_widget(f)
        t.bind(text=l.setter('text'))

        return b

if __name__ == "__main__":
    TutorialApp().run()
```

Checkpoint #1

- รันโปรแกรม
- วาดหน้าต่างของโปรแกรม
- สรุปวิธีสร้าง GUI พร้อมยกตัวอย่างโค้ดประกอบ

BoxLayout



สรุปวิธีสร้าง GUI พร้อมยกตัวอย่างโค้ดประกอบ

3. สร้างไฟล์ชื่อ myapp.py และ tutorial.kv โดยมีโค้ดดังด้านล่าง แล้วทดลองรัน

myapp.py

```
from kivy.app import App
from kivy.uix.scatter import Scatter
from kivy.uix.label import Label
from kivy.uix.floatlayout import FloatLayout
from kivy.uix.textinput import TextInput
from kivy.uix.boxlayout import BoxLayout

class ScatterTextWidget(BoxLayout):
    pass

class TutorialApp(App):
    def build(self):
        return ScatterTextWidget()

if __name__ == "__main__":
    TutorialApp().run()
```

tutorial.kv

```
<ScatterTextWidget>:
    orientation: 'vertical'
    TextInput:
        id: my_textinput
        font_size: 150
        size_hint_y: None
        height: 200
        text: 'default'
    FloatLayout:
        Scatter:
            Label:
                text: my_textinput.text
                font_size: 150
```

Checkpoint #2

- รันโปรแกรม
- อธิบายความแตกต่างกับโปรแกรมในข้อ 2

- ไฟล์ โค้ดในไฟล์ kv เทียบได้กับส่วนไหนของโปรแกรมเดิม

4. ทดลองสร้างไฟล์ myapp.py และ tutorial.py โดยมีโค้ดดังด้านล่าง แล้วทดลองรัน

myapp.py

```
from kivy.app import App

from kivy.uix.scatter import Scatter
from kivy.uix.label import Label
from kivy.uix.floatlayout import FloatLayout
from kivy.uix.textinput import TextInput
from kivy.uix.boxlayout import BoxLayout

import random

class ScatterTextWidget(BoxLayout):
    def change_label_color(self, *args):
        colour = [random.random() for i in range(3)] + [1]
        label = self.ids['my_label']
        label.color = colour

class TutorialApp(App):
    def build(self):
        return ScatterTextWidget()

if __name__ == "__main__":
    TutorialApp().run()
```

tutorial.kv

```
#:import random random

<ScatterTextWidget>:
    orientation: 'vertical'
    TextInput:
        id: my_textinput
        font_size: 150
        size_hint_y: None
        height: 200
        text: 'default'
        # on_text: my_label.color = [random.random() for i in range(3)] + [1]
        on_text: root.change_label_color()
    FloatLayout:
        Scatter:
            Label:
                id: my_label
                text: my_textinput.text
                font_size: 150
```

- ทดลอง **uncomment** และ **comment** ระหว่างบรรทัดต่อไปนี้ สังเกตผลการทำงาน

```
on_text: my_label.color = [random.random() for i in range(3)] + [1]
```

```
on_text: root.change_label_color()
```

Checkpoint #3

- รันโปรแกรม
- อธิบายความแตกต่างกับโปรแกรมในข้อ 3 และการเปลี่ยน comment ในส่วนของ `on_text`

5. เครื่องคิดเลข สร้างไฟล์ `cal.py` และ `calculator.kv` โดยมีโค้ดดังด้านล่าง (สามารถดาวน์โหลดได้จาก <http://fivedots.coe.psu.ac.th/~thanathip.l/download/python/kivy/codes/work3/>) แล้วทดลองรัน `cal.py`

```
from kivy.app import App
from kivy.ui.boxlayout import BoxLayout

class Calculator(BoxLayout):
    label = ''

    def delete(self, instance):
        self.display.text = instance[:0]

    def dell(self, instance):
        self.display.text = instance[:-1]

    def calc(self, instance):
        try:
            self.display.text = str(eval(instance))
            self.result.text = str(eval(instance))
        except Exception:
            self.display.text = '0'
            self.result.text = 'ERROR'
```

```

class CalculatorApp(App):
    trigger = False
    triggerC = False
    triggerD = False

    def build(self):
        return Calculator()

if __name__ == '__main__':
    CalculatorApp().run()

```

calculator.kv

```

<Calculator>:
    id: calculator
    display: label
    result: labelResult

    BoxLayout:
        orientation: 'vertical'

        Label:
            id: labelResult

            canvas.before:
                Color:
                    rgb: .13, .17, .18, 1
                Rectangle:
                    pos: self.pos
                    size: self.size

            halign: 'center'
            valign: 'center'
            font_size: 40
            text_size: 550, None
            size_hint: 1, .15
            shorten: True
            shorten_from: 'right'

        Label:
            id: label

            canvas.before:
                Color:
                    rgb: .13, .17, .18, 1
                Rectangle:
                    pos: self.pos
                    size: self.size

            halign: 'right'
            valign: 'center'

```

```

font_size: 90
text_size: 550, None
size_hint: 1, .35
shorten: True
shorten_from: 'right'

GridLayout:
    cols: 4
    spacing: 1

    canvas.before:
        Color:
            rgb: .3, .34, .36, 1
        Rectangle:
            pos: self.pos
            size: self.size

    Button:
        text: '('
        font_size: 35
        background_normal: ''
        background_color: .2, .24, .26, 1
        on_press: app.triggerD = True
        on_press:
            if (not app.trigger): label.text += self.text
            if (app.triggerC): app.triggerC = False
            if (app.trigger): label.text += '*' + self.text
            if (app.trigger): app.trigger = False

    Button:
        text: ')'
        font_size: 35
        background_normal: ''
        background_color: .2, .24, .26, 1
        on_press:
            if (app.triggerD): label.text += self.text
            if (app.triggerD): app.triggerD = False

    Button:
        text: '%'
        font_size: 35
        background_normal: ''
        background_color: .2, .24, .26, 1
        on_press:
            if (app.trigger): label.text += self.text
            if (app.trigger): app.triggerC = False
            if (app.trigger): app.trigger = False

    GridLayout:
        rows: 2
        spacing: 1

```



```

Button:
    text: 'C'
    font_size: 30
    background_normal: ''
    background_color: .2, .24, .26, 1
    on_press: calculator.delete(label.text)
    on_press: app.trigger = False
    on_press: app.triggerD = False

Button:
    text: 'CE'
    font_size: 30
    background_normal: ''
    background_color: .2, .24, .26, 1
    on_press: calculator.del1(label.text)

Button:
    text: '1'
    font_size: 35
    background_normal: ''
    background_color: .2, .24, .26, 1
    on_press: app.trigger = True
    on_press:
        if (app.triggerC): label.text = self.text
        else: label.text += self.text
        if (app.triggerC): app.triggerC = False

Button:
    text: '2'
    font_size: 35
    background_normal: ''
    background_color: .2, .24, .26, 1
    on_press: app.trigger = True
    on_press:
        if (app.triggerC): label.text = self.text
        else: label.text += self.text
        if (app.triggerC): app.triggerC = False

Button:
    text: '3'
    font_size: 35
    background_normal: ''
    background_color: .2, .24, .26, 1
    on_press: app.trigger = True
    on_press:
        if (app.triggerC): label.text = self.text
        else: label.text += self.text
        if (app.triggerC): app.triggerC = False

Button:
    text: '-'
    font_size: 35

```

```

background_normal: ''
background_color: .2, .24, .26, 1
on_press:
    if (app.trigger): label.text += self.text
    if (app.trigger): app.triggerC = False
    if (app.trigger): app.trigger = False

Button:
    text: '0'
    font_size: 35
    background_normal: ''
    background_color: .2, .24, .26, 1
    on_press: app.trigger = True
    on_press:
        if (app.triggerC): label.text = self.text
        else: label.text += self.text
        if (app.triggerC): app.triggerC = False

Button:
    text: '.'
    font_size: 35
    background_normal: ''
    background_color: .2, .24, .26, 1
    on_press:
        if (app.trigger): label.text += self.text
        if (app.trigger): app.triggerC = False
        if (app.trigger): app.trigger = False

Button:
    text: '='
    font_size: 35
    background_normal: ''
    background_color: .2, .24, .26, 1
    on_press:
        if (app.trigger): calculator.calc(label.text)
        if (app.trigger): calculator.calc(labelResult.text)
        if (app.trigger): app.triggerC = True

Button:
    text: '+'
    font_size: 35
    background_normal: ''
    background_color: .2, .24, .26, 1
    on_press:
        if (app.trigger): label.text += self.text
        if (app.trigger): app.triggerC = False
        if (app.trigger): app.trigger = False

```

Checkpoint #3

- ทำให้เป็นเครื่องคิดเลขที่สมบูรณ์ สามารถทำงานได้ถูกต้อง

