The kv-language:

In a kv-file the so called “kv-language” is used which helps you design layouts for your gui. In general, the language works as follows: you type the widget name followed by a “:” (e.g. “BoxLayout:”) and an intent in the next line. Then you can specify the attributes like height, position etc. of the widget by naming the attribute followed by a “:” and the value (e.g. “height: 40). Furthermore, if you add another widget while being in the indentation of a widget the added widget will be the child-widget. You can also specify the layout of your custom widgets that you use in your app. To declare that you are describing the design of your custom widget you have to mark it with “ <>” around the widget name followed by a “:” (e.g. “<YourCustomClass>:”)

```
38  <OneArrangementItem>:
39      size_hint_y: None
40      orientation: 'horizontal'
41      Label:
42          text: root.exercise_name
43          height: 40
44          size_hint_y: None
45      Button:
46          text: 'up'
47          height: 40
48          size_hint_y: None
49          on_release: root.on_up_button()
50      Button:
51          text: 'down'
52          height: 40
53          size_hint_y: None
54          on_release: root.on_down_button()
55      Button:
56          text: 'remove'
57          height: 40
58          size_hint_y: None
59          on_release: root.on_remove_button()
```

Figure 1

To make things clearer I will describe the example in fig 1: In there the custom class (widget) with the name “OneArrangementItem” is described. OneArrangementItem inherits from “BoxLayout” which you cannot see in the figure 1 but you know if you take a look in utils_gui.py. BoxLayout is a class provided by the framework kivy and in line 39-40 we set the size_hint_y to None (which means the widget we are creating will expand in y-direction as much as it has space) and we set the orientation to horizontal (which means that objects added to the BoxLayout will be added horizontally). Then we add a Label (kivy-class) in our BoxLayout and set the text, height, size_hint_y of that label. Then we add 3 buttons to the BoxLayout. Worth noting is that we bind the buttons to methods of the class OneArrangementItem. “root” is a keyword which always refers to the class we are currently describing so in this case OneArrangementItem. Now whenever we create an instance of OneArrangementItem in our app kivy will use the layout we just described.