Python - Kivy : Utilisez les widget Boutons, Labels et TextInput

Présentation

Voici une activité pour bien comprendre comment :

- créer, positionner et utiliser des boutons, labels et inputs,
- Gérer une liste de boutons
- Comment insérer et utiliser un bouton, un label et un input.

Créer, positionner et utiliser des boutons, labels et inputs

- Crée un dossier **Menu** ;
- Dans ce dossier crée un fichier **main.py** avec le contenu suivant :

main.py

```
from kivy.app import App
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 Ecran(BoxLayout):
   def build(self):
        self.orientation='vertical'
        self.spacing=20
        self.Un_Bouton()
        self.Un_Label()
        self.Un_Input()
   def Un_Bouton(self):
        #On cree un bouton:
        self.Bouton1=Button()
        #On lui donne des proprietes:
        self.Bouton1.text='Clique sur ce bouton pour changer le texte'
        #Une taille en pourcentages:
        self.Bouton1.size hint=(0.5, 0.15)
        #Une position:
        self.Bouton1.pos_hint={'right': 0.75}
        #Une couleur de fond:
        self.Bouton1.background_color=[0,1,0,1]
        #On l'associe a une fonction:
        self.Bouton1.bind(on_press=self.Une_Fonction_Bouton)
        #On l'ajoute au layout principal:
        self.add_widget(self.Bouton1)
    def Un Label(self):
        #On cree un label avec toutes ses proprietes:
        self.Label1=Label(text="Texte du label",font_size=30, color=[1,0,0,1])
        #On l'ajoute au layout principal:
        self.add_widget(self.Label1)
   def Un Input(self):
        #On cree un Input avec des ses proprietes:
        self.Input1 = TextInput(text="Texte initial... à changer", font size=30)
        #On ajoute des proprietes (une largeur de 50% et une hauteur de 20%)
        self.Input1.size_hint_x=0.5
        self.Input1.size_hint_y=0.2
        #On l'ajoute au layout principal:
        self.add widget(self.Input1)
   def Une_Fonction_Bouton(self,instance):
```

```
#On peut changer le texte du bouton:
    #la variable "instance" correspond au bouton qui a ete presse
    instance.text="Bravo ! Tu as cliqué sur moi"
    #On peut changer le texte du label:
    self.Labell.text=self.Inputl.text
    # on reinitialiser le contenu du Input
    self.Inputl.text="Texte initial... à changer"

class Menu(App):
    def build(self):
        Layout=Ecran()
        Layout.build()
        return Layout

if __name__ == '__main__':
    Menu().run()
```

Modifie ce programme pour permettre d'afficher la composition d'un menu :

- en saisissant un élément du menu
- en rajoutant le nouvel élément quand on clique sur un bouton

On ne doit pouvoir saisir qu'un maximum de 5 éléments :

- afficher le nombre d'éléments saisis
- afficher C'est fini. quand 5 éléments ont été saisi.

Bouton

- Crée un dossier **Boutons** ;
- Dans ce dossier crée un fichier main.py avec le contenu suivant (Ce programme affiche une liste de boutons) :

main.py

```
# -*- coding: utf-8 -*-
from kivy.app import App
from kivy.uix.boxlayout import BoxLayout
from kivy.uix.button import Button
class Ecran(BoxLayout):
    def build(self):
        self.orientation='vertical'
        self.spacing=20
        self.Mes_Boutons()
    def Mes_Boutons(self):
        #On cree une liste pour les boutons:
        self.Liste_Boutons=[]
        #On fait tourner une boucle pour creer 10 boutons:
        for i in range(0,10):
            #On ajoute un bouton dans la liste:
            self.Liste Boutons.append(Button())
            #On lui donne un texte qui depend de i:
            self.Liste_Boutons[i].text="Bouton "+str(i)
            #On lui donne un identite "id" pour le retrouver hors de la liste:
            self.Liste_Boutons[i].id="B"+str(i)
            #On lui associe une fonction:
            self.Liste_Boutons[i].bind(on_press=self.Une_Fonction_Bouton)
            #On ajoute le bouton au layout principal:
            self.add_widget(self.Liste_Boutons[i])
        #On ajoute une couleur de fond au dernier bouton:
        self.Liste_Boutons[9].background_color=[0,1,0,1]
    def Une Fonction Bouton(self,instance):
```

Printed on 2025/09/25 04:17

```
#instance correspond au bouton qui vient d'etre active:
    #On change la couleur des neuf premiers boutons:
    for i in range(0,9):
        self.Liste_Boutons[i].background_color=[0.5,0.5,0.5,1] #En gris
    #On change la couleur du bouton active si ce n'est pas le dernier:
    if instance.id!="B9":
        instance.background_color=[1,0,0,1] #En rouge

class BoutonsApp(App):
    def build(self):
        root=Ecran()
        root.build()
        return root

if __name__ == '__main__':
    BoutonsApp().run()
```

Modifie ce programme pour :

- afficher les boutons comme ceux d'une calculatrice,
- ajoute le bouton + permettant de faire une addition et d'afficher le résultat.

Retour au menu de ma première application

• Première application

From: /- Les cours du BTS SIO

Permanent link:

/doku.php/dev/python/kivy/app1/widget

Last update: 2017/03/16 16:36

