

Python - Kivy : Hello World !

Le programme Hello

- Crée un dossier **hello** ;
- Dans ce dossier crée un fichier **main.py** avec le contenu suivant : <code python main.py> from kivy.app import App from kivy.uix.button import Button

```
class Hello(App):
    def build(self):
        return Button(text='Hello World !')
Hello().run() </code>
```

Exécution du programme

- Lance l'exécution du fichier **main.py** depuis Spyder ou en ligne de commande :

```
python main.py
```

- un fenêtre s'ouvre et contient un bouton cliquable sur lequel est inscrit “**Hello World !**” qui couvre toute la fenêtre

Explication

Pour programmer avec Kivy, tu utilises des objets. Voici une explication des instructions du programme :

- from kivy.app import App
 - cela permet d'**importer la classe de base App** car ton application va être une nouvelle classe qui **hérite** de cette App de base
- **from kivy.uix.button import Button**
 - importation à partir du module uix de l'élément d'interface graphique Button. Les éléments graphiques sont appelés des **Widgets**
 - **class Hello(App):**
 - Déclaration de la classe de base de ton application et définition du nom de l'application
 - **def build(self):**
 - cette fonction permet de définir (construire) la composition de l'interface. Le mot clé self signifie qu'il s'agit d'une fonction qui concerne l'objet (la classe Hello()) lui-même.
 - **return Button(text='Hello World !')**
 - initialisation d'un bouton avec affichage d'un libellé. Ce widget sera le widget principal de l'application retourné par la fonction build().
 - **Hello().run()**
 - initialisation et exécution de ton application

Retour au menu de ma première application

▪ Première application

From:
[/ - Les cours du BTS SIO](#)

Permanent link:
[/doku.php/dev/python/kivy/app1/helloworld](https://doku.php/dev/python/kivy/app1/helloworld)

Last update: **2017/05/04 14:17**

