Boa constructor : Tutoriel - Construire votre première application

Source : Tutoriel fourni dans la documentation de BOA Constructor [1)] et traduit en français par Kleiber Ludovic (http://aspn.activestate.com/ASPN/Python/Cookbook/)

Cette section présente un bref tutoriel. Le but de ce tutoriel est de vous familiariser avec l'environnement de développement Boa constructeur. Ce tutoriel vous guidera pas à pas à travers le processus de construction d'un éditeur de texte simple, appelé Notebook. Après avoir cheminé à travers ce tutoriel, vous en saurez assez pour être productif avec Boa Constructor.

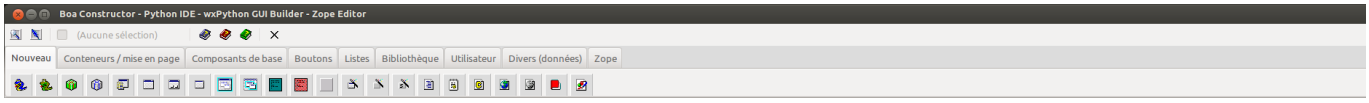
Vous apprendrez à :

- Créer une application.
- Créer des cadres, des menus et des barres d'état.
- Créer des contrôles tels que des boutons, des champs de saisie de texte et des étiquettes.
- Configurer les contrôles suivant vos besoins.
- Travailler avec des boîtes de dialogue classique.
- Concevoir vos propres boîtes de dialogue.

Création d'une nouvelle application

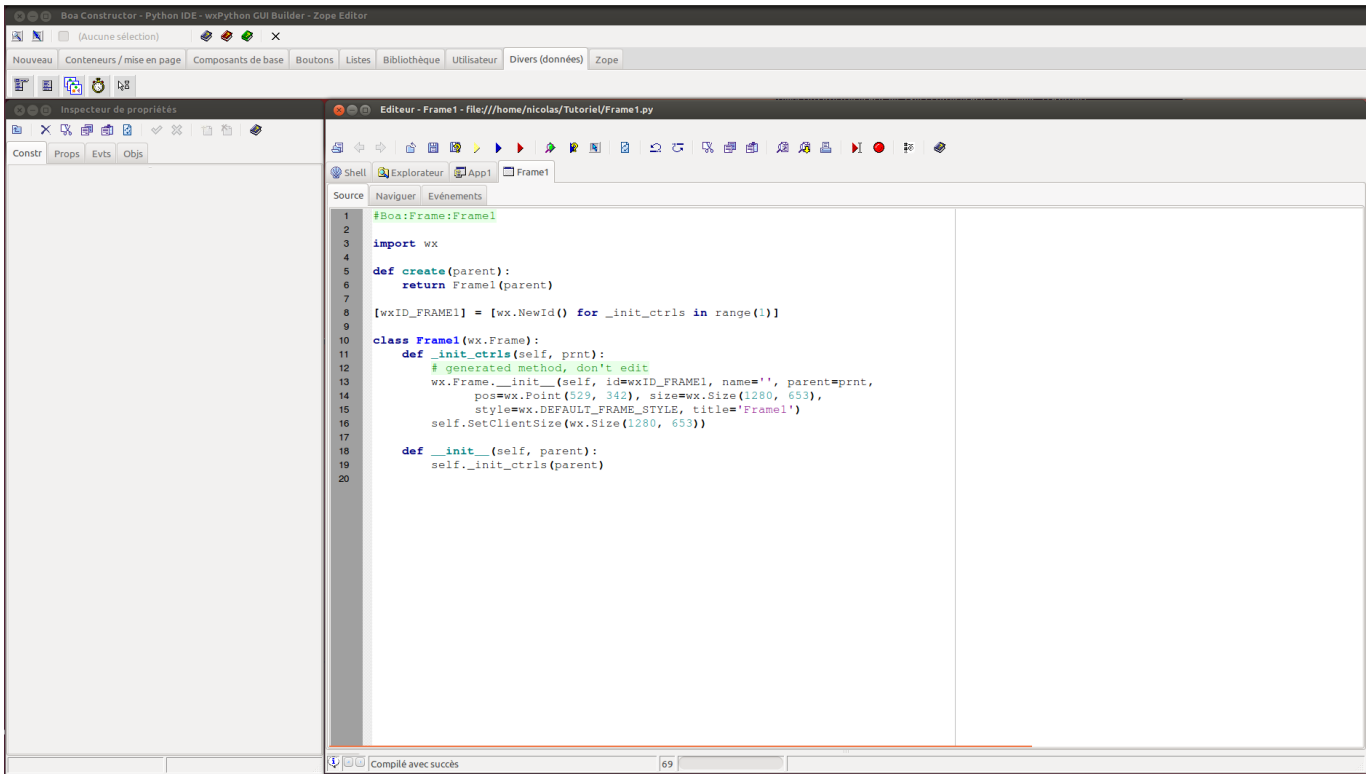
- Décider d'un répertoire destiné à contenir l'application. Si nécessaire, créez le répertoire.
- Créez une nouvelle application en utilisant le bouton ci-dessous nouvelle application à partir de la palette.

Bouton de l'application (info-bulle = wx.APP) : 



Enregistrer le fichier de App1.py et le fichier Frame1.py dans le répertoire que vous avez créé plus tôt. Vous pouvez utiliser le bouton 'Enregistrer' dans la barre d'outils Éditeur. Notez que les astérisques (*) disparaissent à partir du nom quand il est enregistrer. Ceux-ci indiquent qu'il y a des modifications non enregistrées dans le fichier.

Vous avez maintenant une application, qui montre juste un cadre vide. Utilisez le bouton «Démarrer l'application“ sur la barre d'outils de l'Éditeur pour exécuter l'application.



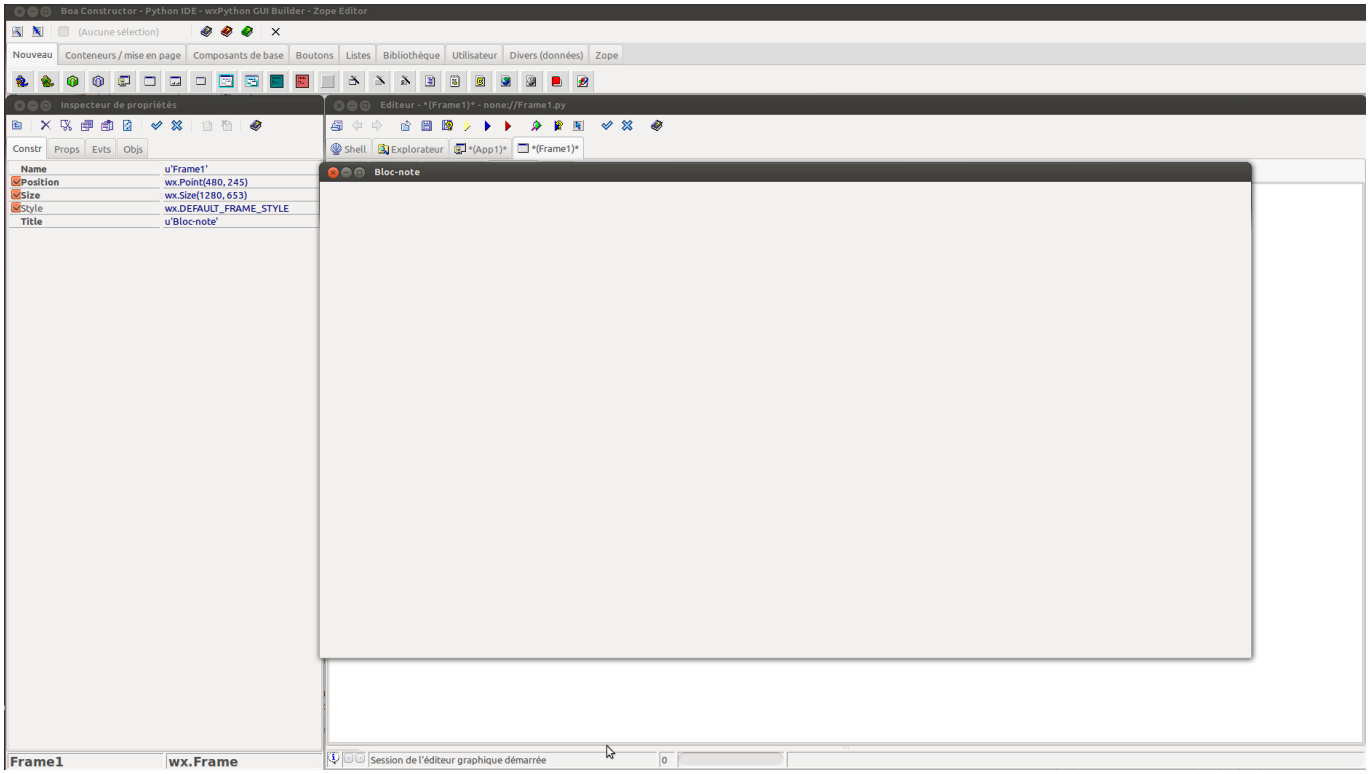
L'image ci-dessus vous montre, dans la section Éditeur, les deux nouveaux fichiers que vous avez créés et enregistrés.

En cliquant sur le bouton de lancement (jaune) vous verrez le résultat de “votre programmation”, c'est à dire juste un cadre vide.

Utilisation de l'Éditeur graphique pour définir le titre

Sélectionnez l'onglet Frame1 dans l'éditeur pour que ce cadre soit édité.

Démarrez l'Éditeur graphique, en cliquant sur le bouton  de la barre d'outils de l’Éditeur.



Le cadre sera affiché comme une zone de dessin (d'abord le titre affiche “Frame1 '

Deux nouveaux onglets seront affichés dans l'éditeur nommé 'Données' et 'Sizers '.

La fenêtre Inspecteur de propriétés affiche le le volet 'const' c'est à dire constructeur. Ce volet vous permet de modifier la taille, la position, le style, le nom de la variable et le titre d'un composant.

Modifiez le champ appelé «Titre». Donner le nom «Bloc-note» au cadre. Lorsque vous appuyez sur la touche retour, vous verrez que le titre de la fenêtre de l’éditeur graphique a changé.

Vous devez enregistrer les modifications en utilisant le bouton 'Post' . Vous pouvez appuyer soit sur le bouton 'Post' de la barre d'outils de l’éditeur ou celui la barre d'outils de l'Inspecteur de propriétés.

La fenêtre de l’éditeur graphique se ferme.

Vous remarquerez que le code source a été mis à jour afin de mettre à jour le titre.

L'éditeur indique que le code source est modifié par des astérisques sur les onglets Frame1, qui demande l'appui sur le bouton Enregistrer.

Ajout d'une barre d'état

Le premier composant que nous allons ajouter à l'application sera une barre d'état. Une barre d'état est utilisé pour donner des informations sur un programme quand il s'exécute. Nous allons utiliser la barre d'état pour indiquer à l'utilisateur ce qui se passe lorsque des actions lentes se produisent comme, donner des messages d'aide simples ou d'autres informations que vous voudriez peut-être montrer.

Sélectionnez l'onglet Frame1 dans l'éditeur pour vous assurez que celui-ci soit édité.

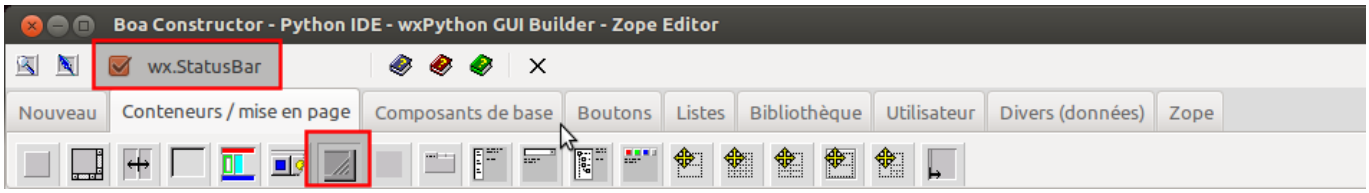
Démarrez l'éditeur graphique, en cliquant sur le bouton dans la barre d'outils de l'éditeur.

Le cadre sera affiché comme une zone de dessin.

Sur la palette, sélectionnez l'onglet intitulé «Conteneurs / Mise en page». Cet onglet contient les entrées pour les composants qui sont utilisés avec les cadres, la barre d'état est l'un d'entre eux.

Déplacez la souris sur les boutons. Des bulles d'aide apparaissent, l'un de ces boutons représente le contrôle wx.StatusBar. Il s'agit du contrôle que nous voulons. Cliquez sur ce bouton en utilisant la souris.

Le bouton devrait se modifier pour indiquer qu'il est pressé. La palette contient une case à cocher pour afficher le type de composant actuellement sélectionné. Celle-ci doit indiquer wx.StatusBar.

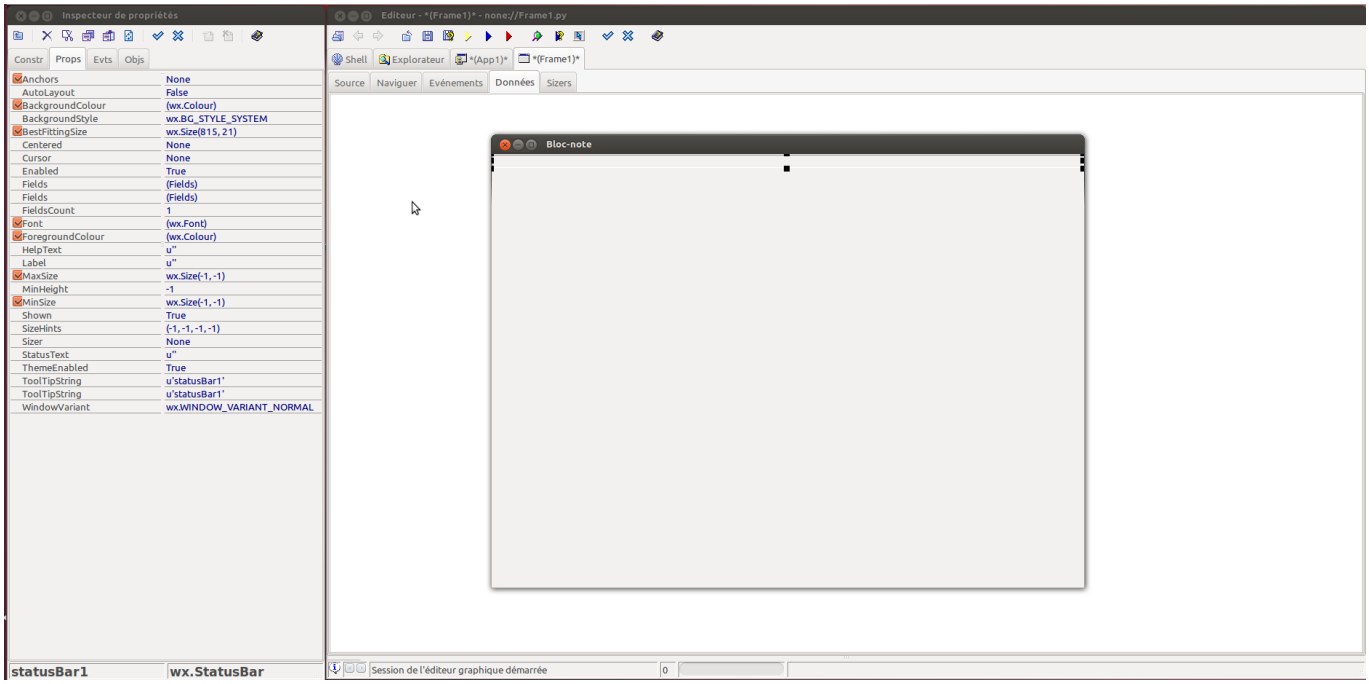


Maintenant, déplacez le curseur de la souris sur le dessin du cadre. Appuyez sur le bouton gauche de la souris dans la zone de dessin. Cela va créer une barre d'état dans le cadre.

La ligne de la barre d'état de l'Inspecteur de propriétés affichera sur la gauche le nom 'statusBar1' du widget actuelle, et sur la droite il montre de quelle classe wxWidget il est dérivé, c'est à dire « wx.StatusBar ».

Dans l'Inspecteur de propriétés sélectionnez l'onglet 'Props'. Cet onglet est utilisé pour configurer les propriétés de notre barre d'état.

Cliquer pour éditer la valeur de 'Fields'. Le champ affichera un bouton marqué "+ + +" au bout de la ligne. Cliquez sur le bouton. Cela ouvre la boite de dialogue de l'«Éditeur de collection».

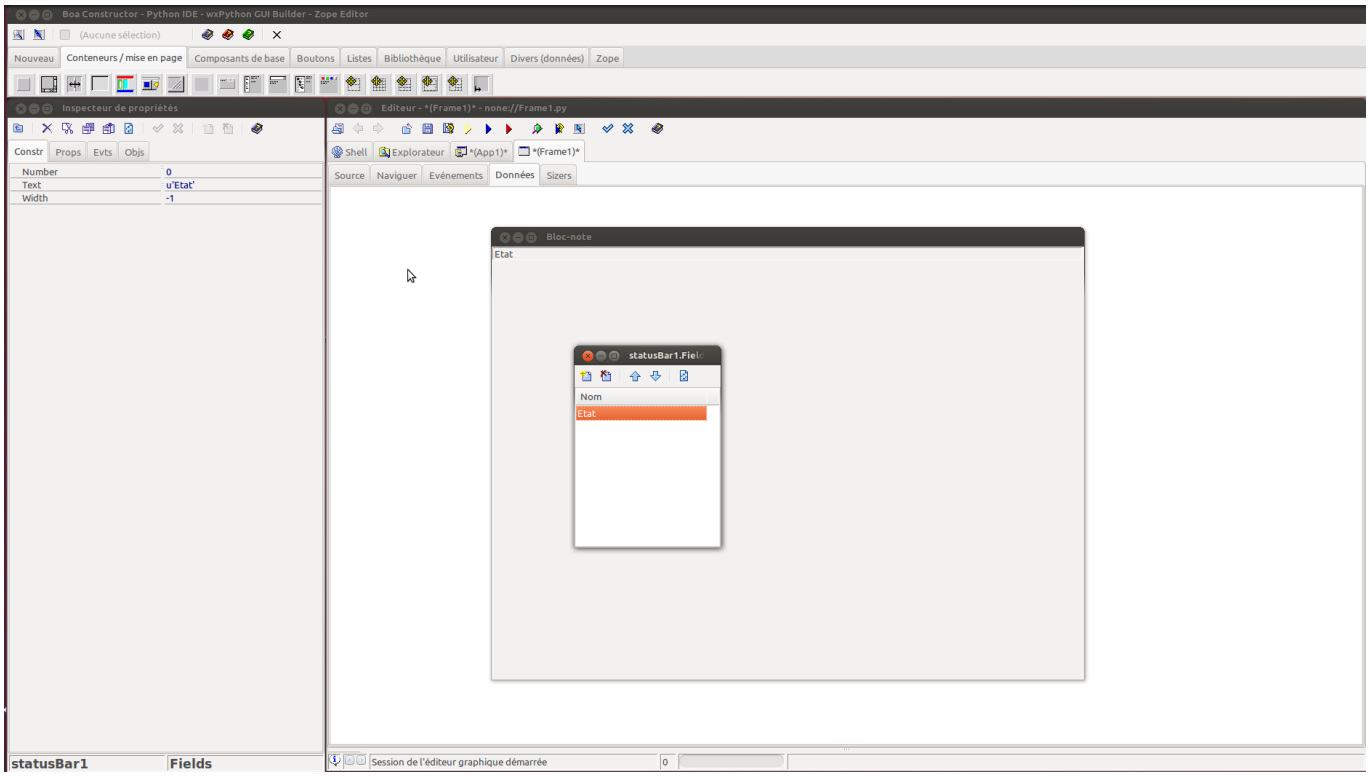


L'éditeur de collection est un outil qui est utilisé le cas échéant pour ajouter plusieurs sous-composants à des composants. Nous allons ajouter un champ à la barre d'état, mais vous pouvez en ajouter de multiples.

Appuyez sur la bouton “Nouveau”  de la Barre d'outils de l'éditeur de collection. Cela crée un nouveau champ dans la barre d'état. Celui-ci devient le champ sélectionné dans l'inspecteur de propriétés.

Modifier le texte de ce champ. Changez le nom de «Fields0» en «Etat».

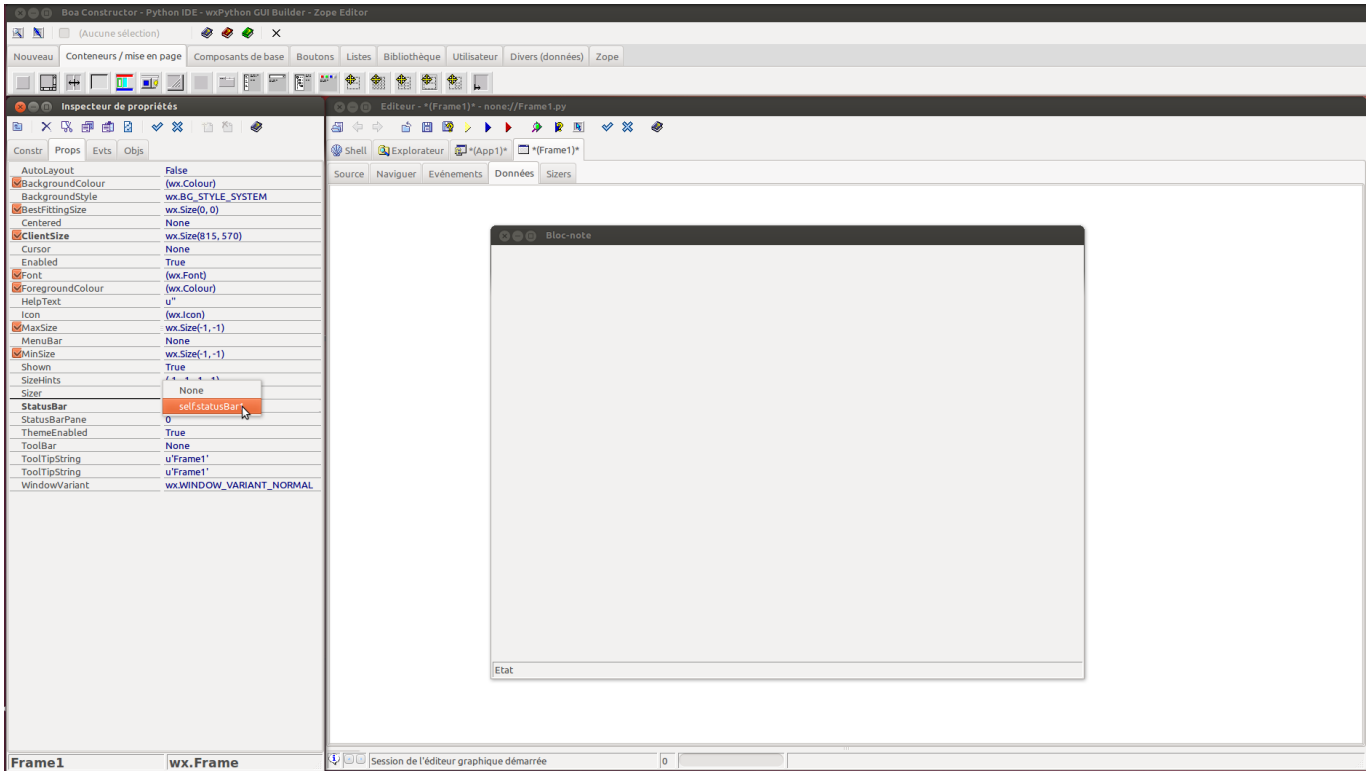
La barre d'outils de l'Éditeur de collection contient un bouton 'Rafraîchir'  . Appuyez sur ce bouton pour voir le changement de l'inspecteur de propriétés dans la fenêtre de l'éditeur de Collection.



Fermez la fenêtre de l'Éditeur de collections. Sélectionnez la fenêtre de l'Éditeur graphique. Cliquez sur la souris n'importe où dans la zone de dessin pour sélectionner le cadre (Frame1) comme composant courant.

Sélectionnez l'onglet props dans l'Inspecteur de propriétés.

Éditer les propriétés “StatusBar” en cliquant au bout de la ligne. Le menu déroulant affiche de notre barre nouvelle d'état. Sélectionnez la, ceci est nécessaire pour que le cadre puisse gérer la barre d'état, c'est à dire qu'il la positionne au bas de l'écran et qu'il la dimensionne



Mettre à jour le code source avec ses modifications en utilisant à nouveau le bouton post .

Enregistrer les modifications du code source en utilisant le bouton Enregistrer dans la barre d'outils de l'Éditeur.

Ajouter une barre de menus

Le composant suivant, que nous allons ajouter à l'application est une barre de menus. Une barre de menu est un composant commun dans les fenêtres des programmes. Notre barre de menu contient deux entrées, fichiers et aide. La sélection de l'une d'elles affichera un menu déroulant. L'utilisateur pourra sélectionner une option dans ce menu déroulant.

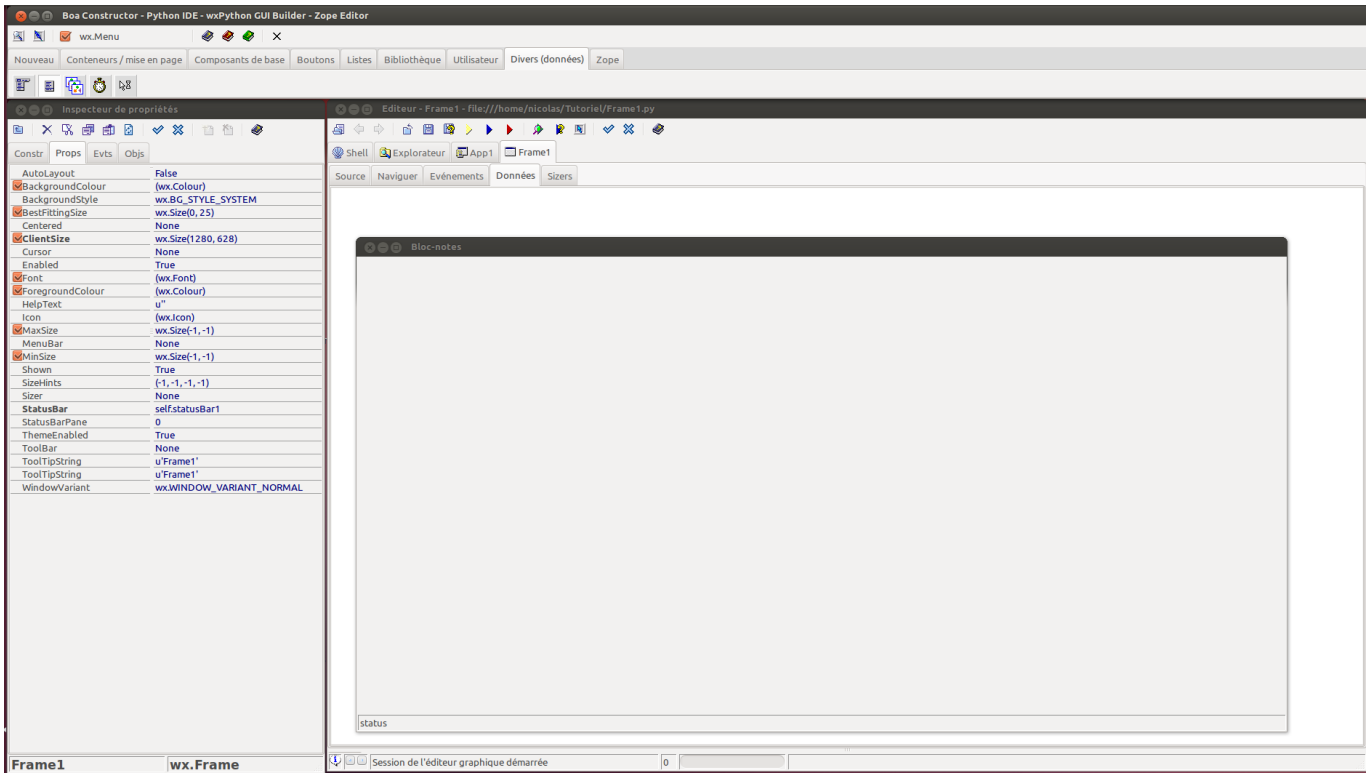
Sélectionnez l'onglet Frame1 dans l'éditeur et vérifier que le cadre soit édité.

Démarrez l'Éditeur graphique, en cliquant sur le bouton Éditeur graphique dans la barre d'outils de l'Éditeur.

L'éditeur graphique va ajouter dans la fenêtre de l'éditeur deux onglets supplémentaires, vous voyez donc les onglets «données» et «Sizers» apparaître. Sur la palette, sélectionnez l'onglet appelé «Utilitaires (données). Le menu déroulant (wx.Menu) est l'un des éléments énumérés dans cet onglet.

Déplacez la souris sur les boutons. Les bulles d'aide montrent que l'un de ces boutons s'appelle un contrôle wx.Menu. Il s'agit du contrôle que nous voulons. Cliquez sur ce bouton.

Le bouton devrait changer de forme pour indiquer qu'il est pressé. La palette contient une case à cocher qui afficher le type de composant actuellement sélectionné. Celle-ci devrait marquer wx.Menu.



Maintenant cliquez sur le bouton gauche de la souris soit dans la vue de l'éditeur sur données soit dans l'Éditeur graphique, dans ce dernier cas, vous devez être prudent afin de vous assurer que vous cliquez sur la zone « Frame1» et non sur la «barre d'état».

Le menu ne sera pas visible sur le cadre. Cependant, il y aura désormais une entrée qui s'affiche dans l'onglet données.

Répétez la procédure. Vous devriez maintenant avoir deux composants wx.Menu visiblement dans données, appelés menu1 et menu2. Sélectionnez menu1 à l'aide de la souris. L'inspecteur de propriétés va maintenant afficher le nom et le titre de menu1.

Modifier le nom 'Name' du premier composant wx.Menu en l'appelant menuFichier. Appelez le seconde composant wx.Menu, menuAide. Définissez respectivement les titres 'Title' Fichier et Aide.

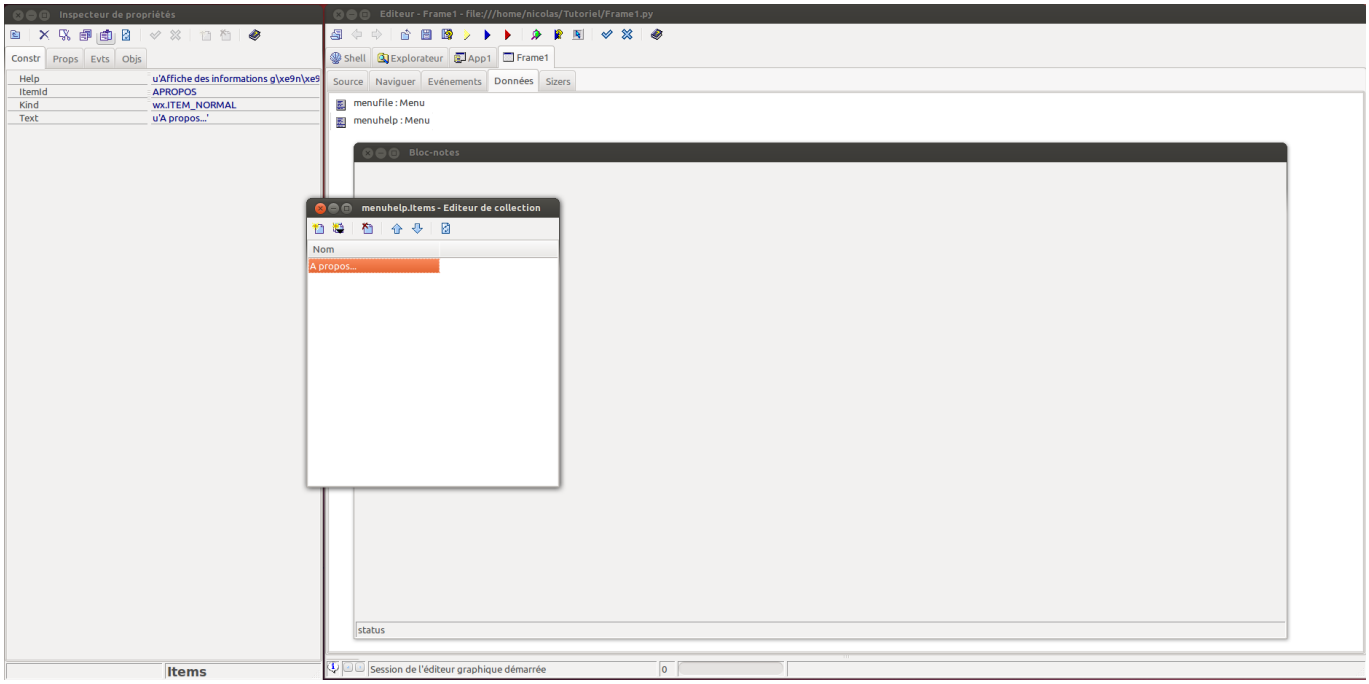
Double-cliquez sur l'entrée menuAide dans l'onglet données. Cela ouvre l' «Éditeur de collection». L'éditeur de collection est utilisée pour ajouter des éléments à nos menus.

Appuyez sur le bouton 'Nouveau' dans l'éditeur de collection. Cela crée un nouvel élément de menu dans le menu déroulant. Celui-ci devient l'élément courant de l'inspecteur de propriétés.

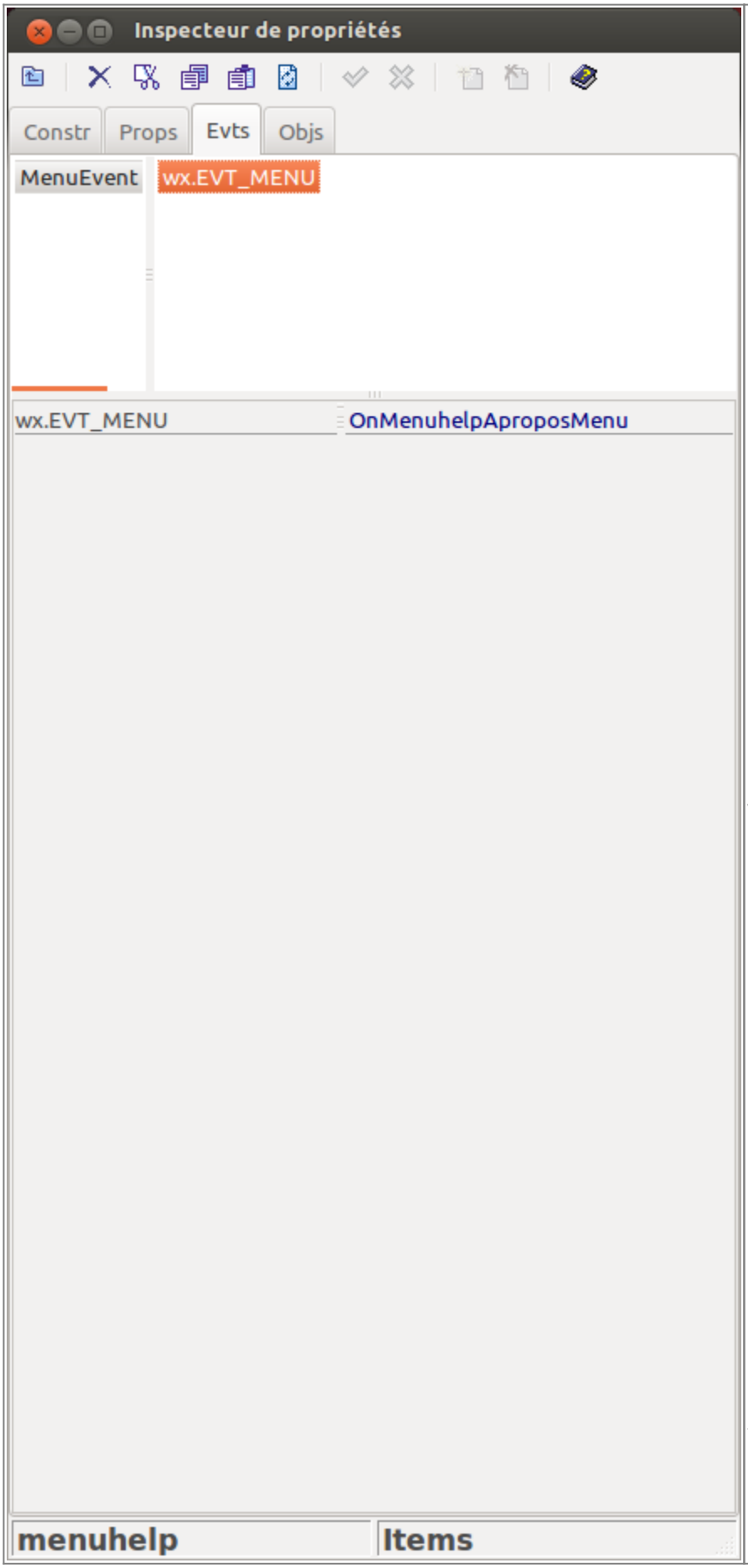
Modifier l'onglet «Constr». Remplacer le nom ('Text') de ' Items0 'par ' A propos de' et je vous recommande également de compléter l'ItemId par 'A_PROPOS_DE'.

Modifier le champ 'Help ' par « Afficher des informations générales sur le bloc-note ».

La barre d'outils de l'Éditeur de collection contient un bouton 'Rafraîchir'  . Appuyez sur ce bouton pour voir les changements de l'inspecteur de propriétés dans la fenêtre de l'éditeur Collection.



Dans l'inspecteur de propriétés, sélectionnez l'onglet Événements ('Evts'). C'est le volet utilisé pour configurer des événements. Nous avons besoin de configurer l'action qui se produit lorsque l'élément de menu 'A propos de' est sélectionné. Lorsque l'élément de menu 'A propos de' est sélectionné, l'événement appelé «EVT_MENU ' est généré et envoyé à notre programme. Nous devons ajouter une méthode à notre classe pour gérer cet événement.



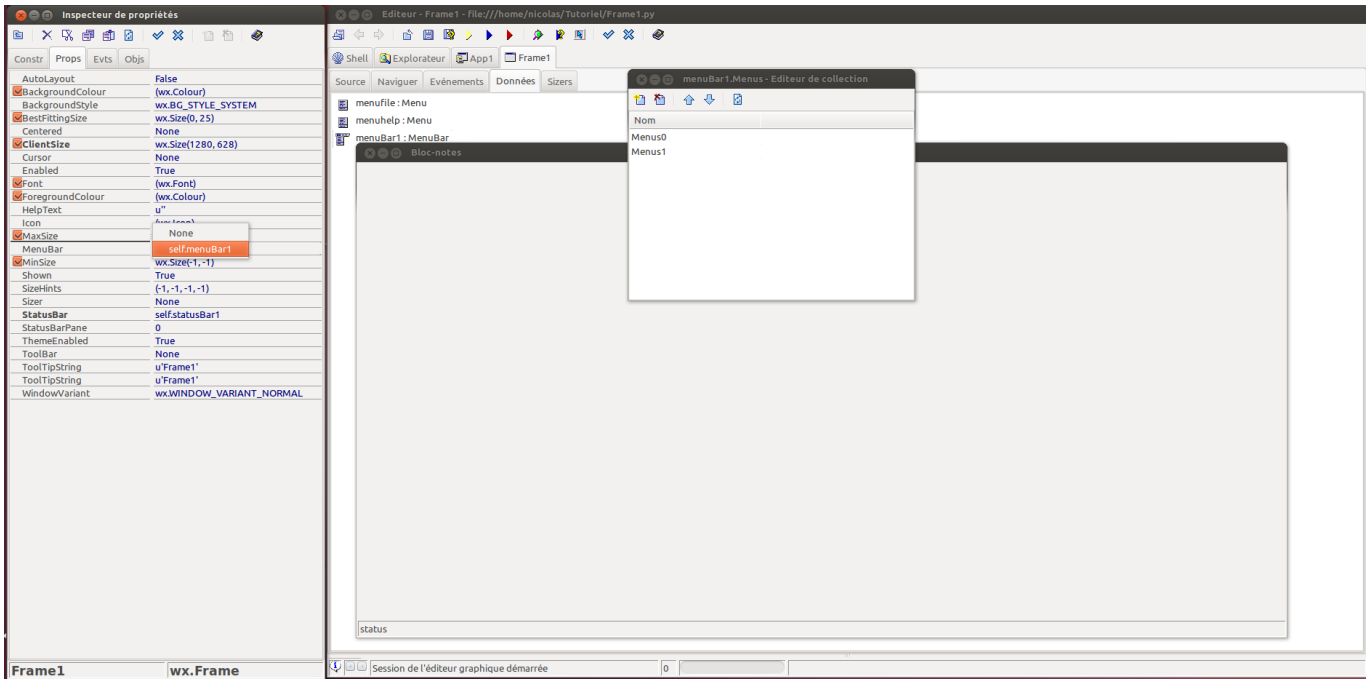
Le côté gauche de la fenêtre des événements montre les groupes d'événements qui sont disponibles.
Pour l'élément de menu, il n'y a qu'un groupe d'événement «MenuEvent ».
Sélectionnez ce groupe en utilisant la souris.
Le côté droit de la fenêtre des événements montre à présent les événements dans le groupe sélectionné.
Pour l'élément de menu du groupes « MenuEvent » sélectionné, il n'y a qu'un seul événement 'EVT_MENU'.
Double Cliquer sur cet événement à l'aide de la souris.
La partie inférieure du volet Événements montre les sous programmes gestionnaires d'événements dans votre application pour le composant courant (ici l'élément de menu 'A propos').
Vous devriez maintenant avoir un nouveau sous-programme gestionnaire appelé OnMenuAideA_propos_deMenu.
Il s'agit du nom de la méthode qui sera invoquée lorsque l'option 'A propos de' est sélectionnée dans le menu Aide.
Noter le nom donné au sous programme gestionnaire d'événements.
Boa Constructor génère les noms de cette manière.
L'événement (Menu) est la dernière partie du nom.
Le composant est la partie du milieu comme ici 'A_propos_de', le sous-composant du composant «menuAide».
Enfin, Boa Constructor suit la convention qui consiste à préfixer tous les gestionnaires d'événements avec le mot «On».
Fermez l'éditeur de collection.

Maintenant, nous devons répéter le processus pour ajouter des options dans le menu fichier.

- Depuis l'onglet données dans l'éditeur, double-cliquer sur l'élément «menuFichier 'pour ouvrir l'éditeur de collection.
- Ajouter cinq nouveaux items.
- Sélectionnez chaque élément de ce menu tour à tour, et nommez les 'Ouvrir', 'Enregistrer', 'Enregistrer sous', 'Fermer' et 'Quitter', et de même que précédemment, je vous recommande de changer le ItemId.
- Entrez du texte d'aide pour chaque élément du menu. Appuyez sur le bouton 'Rafraîchir'  de l'éditeur de collections pour afficher les nouvelles étiquettes.
- Sélectionnez chaque élément du menu tour à tour. Pour chaque élément sélectionnez l'onglet Evénements ('Evts') de l'inspecteur de propriétés, et ajouter un gestionnaire d'événements à chaque élément EVT_MENU.
- Fermez l'éditeur de collection.

Maintenant, nous allons créer la barre de menus.

- Sur la fenêtre de la palette, sélectionnez onglet 'Utilitaires (données). Sur cet barre d'outil, sélectionner le composant barre de menus (wx.MenuBar).
- Déplacez le curseur sur l'onglet données de l'éditeur graphique. Cliquez sur le bouton gauche de la souris sur cet onglet et ajouter à l'application une barre de menu appelé «menuBar1».
- Double-cliquez sur l'élément menuBar1 pour ouvrir l'Éditeur de Collection (comme vous montre l'image ci-dessous vous pouvez garder plusieurs onglets ouverts).
- Ajouter deux éléments à la barre de menu en utilisant 'nouveau' de l'éditeur de collection. Sélectionnez Menus0, dans l'onglet constructeur (Constr) de l'inspecteur de propriétés, modifier le champ “Menu” en cliquant au bout de la ligne. Il apparaît un sous-menu avec trois étiquettes, le constructeur wx.Menu (), et nos deux menus déroulants, sélectionnez l'élément self.menuFichier et écrire 'Fichier' comme titre (Title). Après rafraîchissement, 'Fichier' apparaît dans le menu du premier menu déroulant de la barre de menus.
- Dans l'éditeur de collections sélectionnez le deuxième élément (Menus1). Répétez le processus pour établir un lien entre Menus1 et le menu déroulant self.menuAide, avec comme titre (Title) «Aide» puis rafraîchir.
- Sélectionnez le cadre principal, Frame1 dans l’éditeur graphique. Le cadre doit maintenant être sélectionné dans l'Inspecteur de propriétés.
- Sélectionnez l'onglet Propriétés (Props) de l'inspecteur de propriétés. Modifier le champ 'MenuBar'. Il s'agit d'un menu déroulant. Sélectionnez votre nouvelle barre de menu self.menuBar1. Cette propriété définit quelle barre de menu est associée au cadre.



- Enregistrez les modifications en utilisant à nouveau le bouton Post puis fermer l'éditeur graphique et laisser Boa générer le code source.
- Enregistrez le code généré dans votre fichier source Frame1.py
- Exécutez le programme.

Vous devriez maintenant voir les menus et la barre d'état.

Lorsque vous sélectionnez une option de menu, le texte d'aide doit apparaître dans la barre d'état.

Ajout de contrôle Texte

La tâche suivante consiste à ajouter le composant principal de notre cadre c'est à dire un éditeur de texte. Ce composant est appelé wx.TextCtrl.

Ouvrez à nouveau l'éditeur graphique pour modifier le cadre, Frame1.py.

Sur la palette, sélectionnez l'onglet des « Composants de base ». Sélectionnez le wx.TextCtrl.

Astuce : Vous pouvez passer le pointeur de la souris sur chaque composant pour connaître son nom.

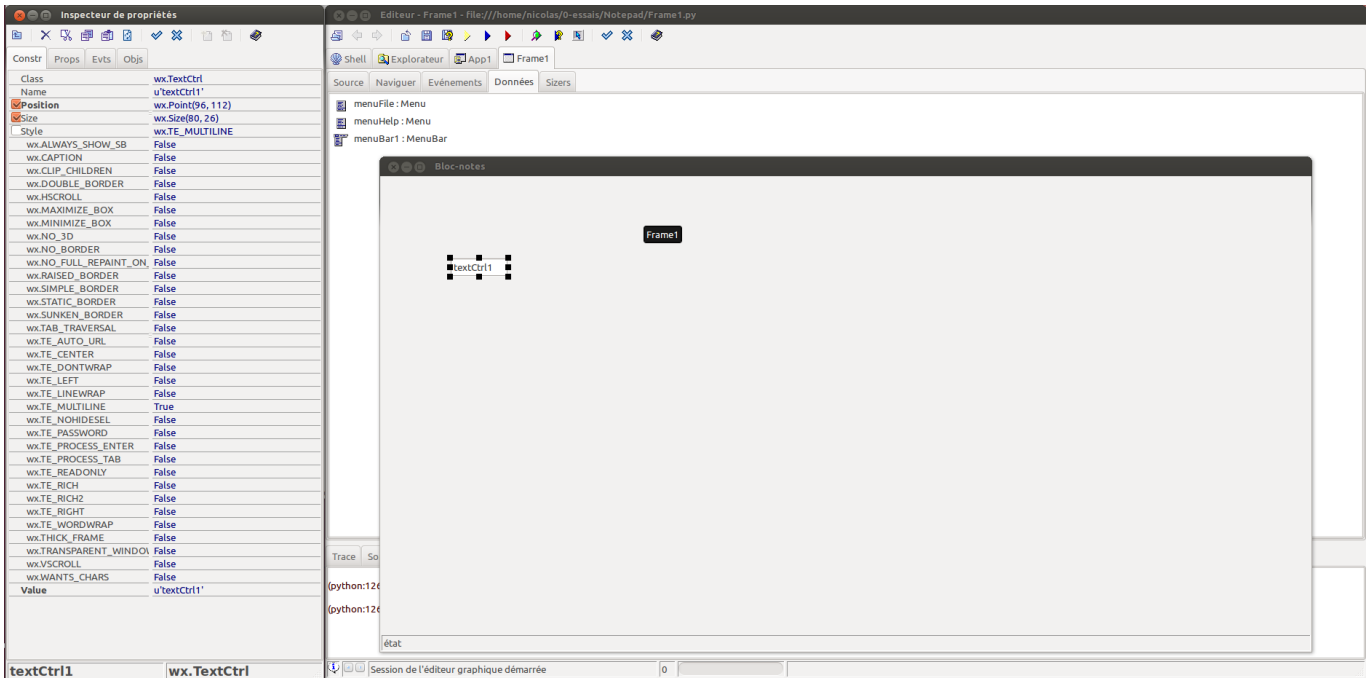
Déplacez le pointeur de votre souris sur la fenêtre de l'éditeur graphique, veiller à ce que l'info-bulle montre 'Frame1', puis cliquez sur le bouton gauche de la souris. Un nouveau composant saisie de texte sera dessiné. Nous n'avons pas déterminé la taille du composant. Par défaut, il va remplir toute la surface disponible, c'est à dire entre la barre d'état et la barre de menu.

La valeur par défaut de wx.TextCtrl est d'une seule ligne. Nous devons indiquer à la commande, que nous voulons que l'entrée des données du composant se fasse sur plusieurs ligne. Pour cela nous éditons le champ «style» dans l'inspecteur de propriétés. C'est sur l'onglet constructeur (Constr).

Modifier le style et le définir à wx.TE_MULTILINE. Vous pouvez saisir cette valeur dans le champ ou cliquer sur la case à cocher à gauche du champ 'style' et Boa vous montrera tous les styles disponibles. En cliquant dessus, changer à «Vrai» (True) les styles que vous voulez utiliser.

Le champ style contient alors le code python validé. Pour définir deux styles logiques vous devez les séparer par un '|'. Vous pouvez voir tous les styles disponibles pour la wx.TextCtrl dans l'aide en ligne de la classe wx.TextControl de wxPython.

Astuce : Utilisez Ctrl-H et entrez “wxtextctrl» pour obtenir de la documentation et trouver les descriptions des différents styles, notez que certains d'entre eux pourraient ne pas être indiqué dans wx.TextCtrl comme ceux par exemple hérités de wx.Window. Actuellement Ctrl-H ne fonctionne pas dans la fenêtre de l'éditeur graphique, mais à peu près n'importe où ailleurs dans Boa.



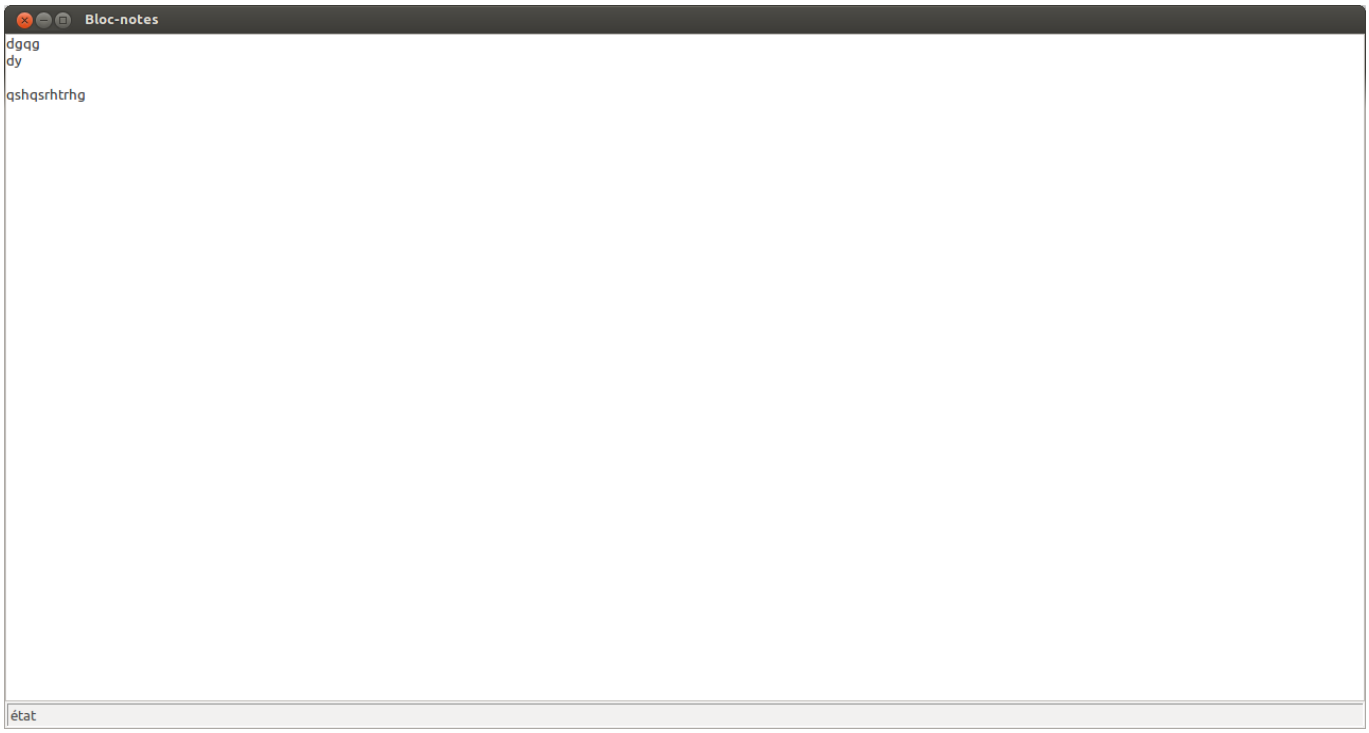
Renommez le champ de texte. Le nom par défaut est 'textCtrl1 ». renommer le en «EditeurDeTexte».

Dans l'onglet constructeur (Constr) il y a un champ appelé «Value». Ce champ contient le texte par défaut affiché dans notre composant éditeur de texte. Effacer le contenu de ce champ.

Mettre à jour le code source avec votre nouveau composant en utilisant à nouveau le bouton Valider

Sauvegardez les modifications du code source.

Exécutez votre application.



Le champ éditeur de texte est automatiquement dimensionnée dans l'espace disponible.

Si vous redimensionnez le cadre, le contrôle est redimensionné.

Notez que wxWidgets vous fournit une barre de défilement. Le champ sera automatiquement ajusté, faites défiler si vous allez au-delà du bas. Si vous tapez une ligne plus longue que la largeur de la fenêtre d'édition, elle s'arrêtera.

Vous avez également les fonctions couper, coller, et de sélection de bloc par défaut.

Ajout de fonctions au menu Fichier

La tâche suivante consiste à interagir avec l'utilisateur pour implémenter les fonctions du menu. Des boîtes de dialogues sont utilisés pour obtenir immédiatement une information de l'utilisateur. Les boîtes de dialogues sont des sortes d'application, où vous ne pouvez pas utiliser les autres fenêtres en cours dans l'application jusqu'à ce que le dialogue soit terminé.

Les boîtes de dialogues sont placés directement dans le code source. Elles ne sont pas placés dans l'éditeur graphique. Elles sont placés avec l'éditeur. Dans le code source de Frame1.py, allez dans le gestionnaire d'événements (onglet Evénements) pour ouvrir l'événement. Cette méthode est appelée OnMenuFichierOuvrirMenu. Double cliquez dessus. Nous allons placer la boîte de dialogue 'Ouvrir Fichier ' dans cette méthode. Placez le curseur du clavier directement au dessus de « event.Skip () ». Nous allons insérer notre nouveau code ici.

Appuyez sur “alt-t” et sélectionnez “wx.FileDialog 'à partir du menu déroulant et Boa Constructor va coller directement le squelette du code de votre méthode dans de gestionnaire d'événements.

Notez le code 'dlg.Destroy ()', qui est très important pour détruire les boîtes de dialogues !

La section du code source doit maintenant se présenter comme suit:

```
def OnMenuFileOpenMenu(self, event):
    dlg = wx.FileDialog(self, "Choose a file", ".", "", "*.*", wx.OPEN)
    try:
        if dlg.ShowModal() == wx.ID_OK:
            filename = dlg.GetPath()
            # Your code
    finally:
        dlg.Destroy()
    event.Skip()
```

Ce squelette de code crée une boîte de dialogue. Elle dialogue avec l'utilisateur. Lorsque le dialogue est terminé, elle est détruite.

Les mots « # Your code » marquent la position où nous pouvons insérer notre propre code. Ce code est déclenchée lorsque la boîte de dialogue retourne un wx.ID_OK, par exemple lorsque l'utilisateur a cliqué sur le bouton 'Ouvrir'. Nous allons insérer notre code ici. Le wx.TextCtrl dispose d'une méthode que nous allons utiliser pour charger un fichier dans la fenêtre d'édition, nous utilisons pour cela la méthode « LoadFile ».

Vous pouvez supprimer le saut « event.Skip () » puisqu'aucun autre événement aura besoin d'être appelé dans ce cas. Le nom de « event.Skip () »est un peu déroutant, vous devez faire appelle à « event.Skip () » dans les événements, lorsque d'autres gestionnaires d'événements doivent également être exécutées.

Il était nécessaire que dans le code généré, Python signale une erreur si il existe une méthode sans corps.

Dans le cadre de la fonction de notre application, nous devons être en mesure d'accéder au nom du fichier de telle sorte que l'option de menu “Enregistrer” puisse enregistrer ce fichier, donc nous avons ajouté la ligne « self.FileName = filename ».

La ligne 'self.SetTitle(('Bloc-note - %s') % filename)' change le titre pour montrer quelle fichier est en cours d'élaboration.

La liste ci-dessous montre notre nouveau code.

```
def OnMenuFileOpenMenu(self, event):
    dlg = wx.FileDialog(self, "Choose a file", ".", "", "*.*", wx.OPEN)
    try:
        if dlg.ShowModal() == wx.ID_OK:
            filename = dlg.GetPath()
            # Your code
            self.textEditor.LoadFile(filename)
            self.FileName=filename
            self.SetTitle(('Notebook - %s') % filename)
    finally:
        dlg.Destroy()
```

Nous devons répéter l'exercice pour implémenter la fonction «Enregistrer sous». Insérer une boîte de dialogue Fichier dans le corps de «OnFichierEnregistrerSousMenu».

Il s'agit d'une boîte de dialogue d'enregistrement de fichier. Changer l'intitulé au deuxième paramètre de wx.FileDialog par “Enregistrer sous”. Changer le style au sixième paramètre par wx.SAVE. Appeler la méthode d'enregistrement du fichier SaveFile.

Encore une fois, nous enregistrons la donnée du nom de fichier utilisée par l'option du menu”Enregistrer”.

La liste ci-dessous montre le code.

```
def OnMenuFileSaveasMenu(self, event):
    dlg = wx.FileDialog(self, "Save file as", ".", "", "*.*", wx.SAVE)
    try:
        if dlg.ShowModal() == wx.ID_OK:
            filename = dlg.GetPath()
            # Your code
            self.textEditor.SaveFile(filename)
            self.FileName=filename
            self.SetTitle(('Notebook - %s') % filename)
    finally:
        dlg.Destroy()
```

Ensuite, nous allons mettre en œuvre l'option «Fermer» du menu. Dans cette méthode, nous avons simplement effacer le contrôle éditeur de texte, la variable membre FileName et réinitialiser le titre

```
def OnMenuFileCloseMenu(self, event):
    self.FileName = None
    self.textEditor.Clear()
    self.SetTitle('Notebook')
```

Ensuite, nous allons implémenter l'option «Quitter» du menu. Dans cette méthode, nous devons mettre fin à l'application. Toutes les demandes wxPython sont terminées par la fermeture de la fenêtre de niveau supérieur. Dans notre cas nous avons seulement la fenêtre Frame1. Pour mettre fin à l'application nous invoquons la méthode Close () pour Frame1.

```
def OnMenuFileExitMenu(self, event):
    self.Close()
```

Ensuite, nous allons implémenter l'élément “Enregistrer” du menu. L'élément enregistrer du menu utilise le nom courant du fichier, qui est stocké dans la variable self.FileName.

Quand il n'y a pas de nom de fichier courant, la variable self.FileName est défini sur Aucun (None). Dans ce cas, l'option de menu “Enregistrer” doit agir en tant que l'option de menu «Enregistrer sous».

La variable FileName doit être créée lors de la création de Frame1. Nous devons l'ajouter à l’éditeur. Vous pouvez ajouter votre code d'application pour la fin du constructeur par défaut (init).

```
def __init__(self, parent):
    self._init_ctrls(parent)
    self.FileName=None
```

Maintenant, nous sommes en sécurité à mettre en œuvre la fonctionnalité Enregistrer. Nous vérifions s'il ya un nom de fichier courant. Si il n'y a que nous pouvons enregistrer le contenu de ce nom de fichier. Sinon, il suffit d'appeler la «Enregistrer sous» méthode.

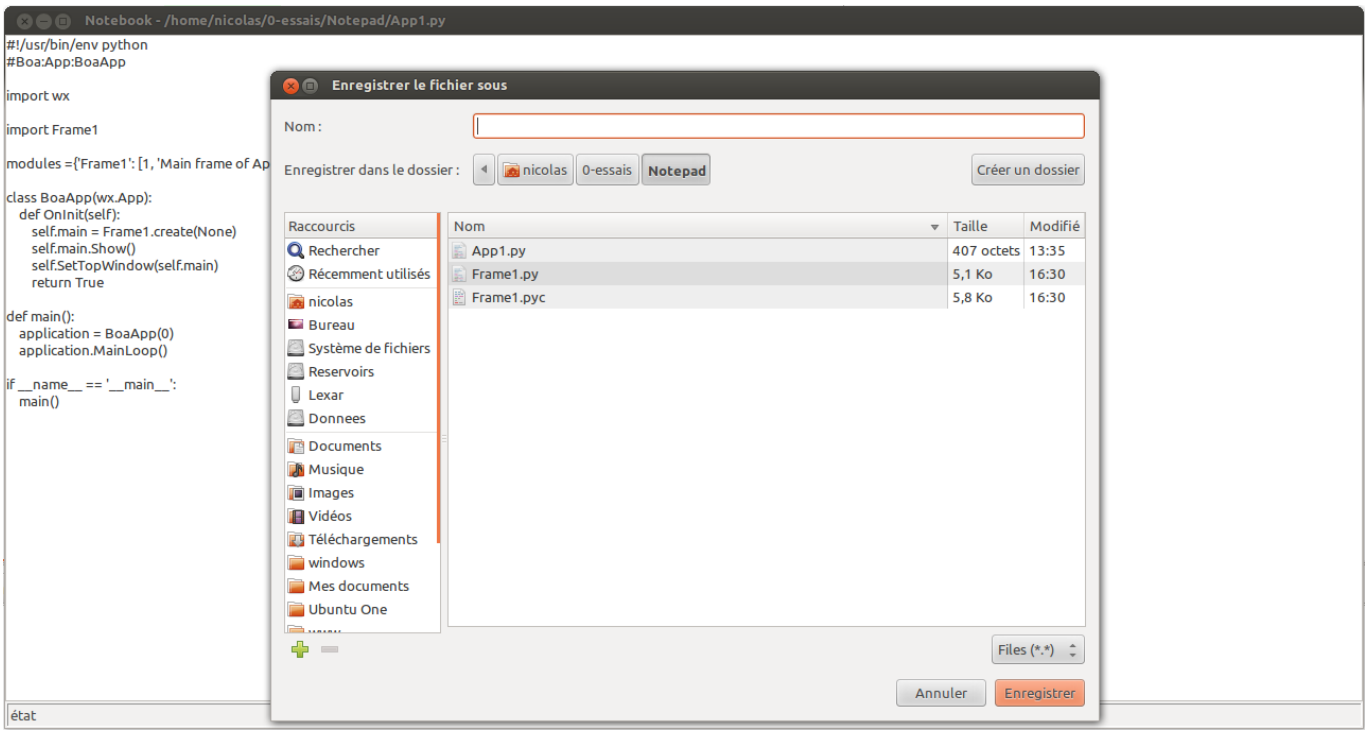
```
def OnMenuFileSaveMenu(self, event):
    if self.FileName == None:
        return self.OnFileSaveasMenu(event)
    else:
        self.textEditor.SaveFile(self.FileName)
```

Nous avons maintenant mis en œuvre la fonctionnalité de l'éditeur. Nous pouvons ouvrir des fichiers, les modifier et les enregistrer.

Votre éditeur devrait ressembler à ce qui est montré sur l'image ci-dessous.

Le fichier a été ouvert App1.py

Puis l'option de menu “Fichier / Enregistrer sous” a été sélectionné



Création d'une boîte de dialogue

Les boîtes de dialogue sont utilisés pour interagir avec l'utilisateur et obtenir des entrées spécifiques. Dans les sections précédentes, nous avons utilisé la boîte de dialogue pré-intégré wx.FileDialog.

Nous allons maintenant développer notre propre boîte de dialogue pour l'option de menu À propos de.

Pour la boîte de dialogue que nous allons créer, il faudra une nouvelle fenêtre. Ce n'est pas un composant de la fenêtre Frame1. Elle existe, dans notre application, dans un fichier Python séparé.

Sélectionnez “App1” le module d'application dans l'éditeur. Choisissez l'onglet « Application ».

Sur la palette, sélectionnez l'onglet 'Nouveau'. Sélectionnez le bouton 'wx.Dialog'. Cela va créer un nouveau fichier source Dialog1.py, et ajouter automatiquement ce nouveau fichier source à votre module d'application.

Sélectionnez l'onglet Frame1. Nous voulons écrire le code pour le menu option 'A propos', qui est utilisé pour afficher la boîte de dialogue. Cette option est mis en œuvre par la méthode «OnHelpAboutMenu ».

Le code est le suivant :

```
def OnMenuHelpAboutMenu(self, event):
    dlg = Dialog1.Dialog1(self)
    try:
        dlg.ShowModal()
    finally:
        dlg.Destroy()
```

Ce code référence le module Dialog1. Avant que ce code fonctionne, il faut importer le module Dialog1. Par convention, nous conservons les importations au début du code source. Ajouter la ligne 'import Dialog1' au fichier Frame1.py après la ligne 'import wx'.

```
import wx
import Dialog1
```

Enregistrer les trois fichiers sources. Vous pouvez exécuter l'application maintenant. Lorsque vous sélectionnez l'option «À propos» du menu 'Aide', votre nouvelle boîte de dialogue apparaît. Notez que la boîte de dialogue est modale, c'est à dire que vous devez la fermer avant que vous puissiez accéder à la fenêtre Frame1. Quittez l'application et revenez à Boa Constructor.

Maintenant, nous allons ajouter des champs à la boîte de dialogue. Pour cet exercice, nous aurons besoin d'un fichier bitmap. Pour la démonstration, j'ai utilisé un appelé Boa.jpg. Vous pouvez créer votre propre image bitmap en utilisant un utilitaire de peinture. Copiez le bitmap à votre répertoire de l'application.

Sélectionnez l'onglet Dialog1.py. Démarrez le l’éditeur graphique en cliquant sur le bouton l’Editeur graphique .

Nous allons d'abord ajouter une étiquette à la boîte de dialogue. Sélectionnez «composants de base» sur la palette. A partir de cet onglet sélectionnez le contrôle wx.StaticText. Dans l’éditeur graphique, cliquer sur le bouton de la souris pour créer le contrôle.

Dans l'inspecteur de propriétés, modifier le paramètre «Label». Saisir la valeur «Notebook - Simple éditeur de texte». Notez que l'étiquette dans l’éditeur graphique va croître pour accueillir votre texte.

Nous utilisons la propriété de style pour configurer l'alignement du texte dans l'étiquette. Saisissez le

paramètre de style «wx.ALIGN_CENTRE» ou sélectionner ce style après avoir cliqué sur la case à cocher à gauche de style.

Sélectionnez l'onglet 'Props dans l'inspecteur Propriétés. Modifiez le champ appelé «police». Définir la police dans une police assez grande, ex. 12 ou 14 points. Notez que vous pouvez changer à la fois la police et la taille de point avec ce paramètre.

Dans la fenêtre Editeur graphique, votre étiquette apparaîtra avec huit poignés sur les bords. Vous cliquez sur le bouton gauche de la souris (et maintenez-la enfoncée) sur l'un de ces marqueurs, puis déplacez la souris pour redimensionner la zone. Vous pouvez également cliquer dans le centre de l'étiquette, et maintenez enfoncé le bouton de la souris, pour déplacer l'étiquette. Positionner l'étiquette au centre de la partie supérieure de la boîte.

Maintenant, ajoutez une autre étiquette en dessous de la première. Définissez le texte comme ceci 'Ceci est ma première application avec Boa Contstructor ». Dans l'Inspecteur de propriétés, sélectionnez onglet 'Props'. Modifier la valeur “BackgroundColour'. Choisissez une couleur dans l'ensemble disponible et appuyez sur OK. Maintenant repositionner et redimensionner votre étiquette jusqu'à ce qu'elle soit équilibrée.

Ensuite, nous allons ajouter le bitmap. D'après Composants de base sélectionnez le contrôle wx.StaticBitmap. Placez le sous la deuxième étiquette de votre boîte de dialogue. Dans l'Inspecteur de propriétés, sélectionner l'onglet Constr. Modifiez le champ Bitmap. Cela vous donnera la boite de dialogue «Ouvrir un fichier». Choisissez l'image que vous avez dessiné au paravent. L'objet wx.StaticBitmap dans l'editeur graphique va changer pour accueillir votre image bitmap. Déplacez l'image bitmap jusqu'à ce qu'elle soit équilibré sous les deux étiquettes.

Enfin, nous allons ajouter un bouton à la boîte de dialogue. Dans la palette sélectionner l'onglet 'Boutons'. Sélectionnez le type de bouton de base, wx.Button. Placez le sous l'image bitmap. Sur l'onglet Constr de l'inspecteur de propriétés modifier «label». Changez le et saisissez «Fermer». Sélectionnez l'onglet Evts dans l'inspecteur de propriétés. Ajouter un gestionnaire pour le type d'événement EVT_BUTTON.

<note tip>sélectionnez d'abord le groupe d'événements, puis l'événement.</note>

Ce sont tous les composants que nous avons ajoutés à la boîte de dialogue. Taille de la boîte de dialogue pour accueillir les contrôles. Repositionner et redimensionner les contrôles jusqu'à ce que vous sentez qu'ils sont bien équilibrés.

Sélectionnez Dialog1 dans l'éditeur graphique. Dans l'onglet Constr de l'Inspecteur de propriétés, modifiez le paramètre « Title ». Pour le changez, saisissez «À propos de Notebook».

Appuyer sur le bouton  pour mettre à jour le code source après vos modifications.

Enfin, nous avons besoin de mettre en œuvre le gestionnaire d'événements pour le bouton Fermer. Dans l'éditeur, sélectionner « source » de Dialog1. Allez au code source de votre méthode 'OnButton1Button'.

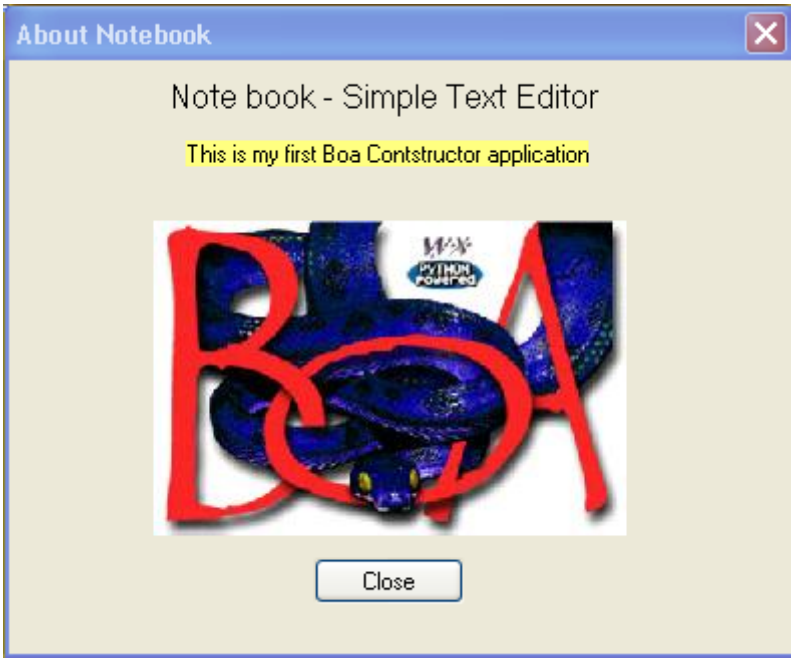
Nous allons utiliser la même méthode «Fermer» que nous avons utilisé dans l'élément du menu “Quitter”. Notez que celle ci ferme la fenêtre. La fermeture des fenêtres issue de l'application racine. Toutefois, la fermeture d'une fenêtre enfant sera tout simplement de revenir à la fenêtre parent.

```
def OnButton1Button(self, event):

```

```
    self.Close()
```

Exécutez l'application. Votre nouvelle boîte de dialogue devrait ressembler à ceci.



Félicitations: Vous avez construit votre première application utilisant Boa Constructor. Votre rédacteur en chef est terminée. Dans ce tutoriel, vous avez utilisé les composants de base de Boa. Prenez le temps de revoir ce que vous avez fait jusqu'à présent. Vous avez appris à:

- Créer une application.
- Créer des cadres, des menus et barres d'état.
- Créer des contrôles tels que des boutons, des champs de saisie de texte et des étiquettes.
- Configurer les contrôles à vos besoins.
- Travailler avec les boîtes de dialogue courantes.
- Concevez vos propres boîtes de dialogue.

Créer une fenêtre d'application à l'aide de l'organisateur (Sizer)

Calibreurs sont une excellente façon de vous assurer que votre mise en page graphique est agréable et propre. Ils viennent particulièrement pratique lorsque vous ne savez pas exactement combien d'espace un contrôle a besoin et / ou devraient être autorisés à utiliser, ce peut être le cas lorsque vous internationaliser votre application (I18N) ou pour les contrôles tels que des listes ou des grilles où vous voulez de donner autant d'espace que possible pour eux (ou peut-être aussi peu que possible).

S'il vous plaît noter que la suit sera de vous expliquer comment utiliser des calibreurs dans Boa (à noter que cela suppose que Boa soit en version 0.6.x). Pour plus d'informations sur les calibreurs vous

devriez consulter la documentation wxPython, la démo wxPython et vous pouvez également (si ce n'est pas assez !) pour comprendre les calibreurs, consulter les liens utiles suivants .

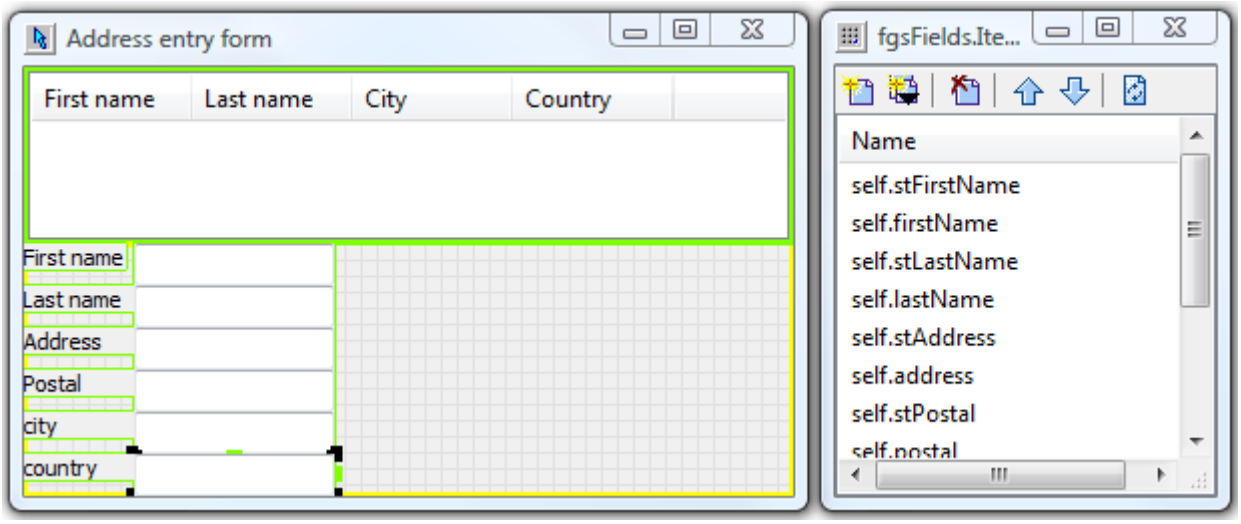
- http://wiki.wxpython.org/index.cgi/UsingSizers
- http://wiki.wxpython.org/index.cgi/LearnSizers1
- http://wiki.wxpython.org/index.cgi/wxDesigner_20Sizer_20Tutorial

Nous allons utiliser un wx.Frame et de créer un écran pour l'entrée des informations d'adresse.

- Fermez tous les fichiers source dans votre éditeur, afin de ne pas ajouter à l'application que vous avez créé précédemment.
- Sur la palette, sélectionnez le 'Nouveau' volet. Sélectionnez le 'wx.Frame bouton. Cela va créer un nouveau fichier source * (Frame1) *.
- Cliquez sur le bouton Enregistrer (ou le menu File / Save) et enregistrez-le sous AddressEntry.py.
- Sélectionnez dans le menu Modifier l'option Ajouter coureur module. Cela va ajouter un peu de code à votre fichier de sorte que vous pouvez l'exécuter sans avoir à disposer d'un fichier wx.App séparé.
- Enregistrez le fichier et vous pouvez exécuter cette application, vous verrez juste Frame1 dans la barre de titre et un fond gris.
- Sélectionnez le volet AddressEntry. Démarrez le Concepteur en cliquant sur le bouton Designer.
- Sur la palette, sélectionnez les “conteneurs / Mise en page” volet. Cliquez sur le bouton wx.Panel pour le sélectionner et cliquez n'importe où dans le cadre AddressEntry. Cela va supprimer le panneau sur votre cadre.
- Sur le volet même palette, cliquez sur le bouton wx.BoxSizer pour le sélectionner et cliquez n'importe où sur le wx.Panel vous venez d'ajouter à votre image. Vous devriez voir une ligne jaune autour de votre panneau.
- Message ces changements et enregistrer le fichier et ré-ouvrir le Concepteur.
- Sur le volet sizer, cliquez sur le boxSizer1 et renommez-le par exemple à bsMain.
- Amener le concepteur à l'avant-plan (par exemple il suffit de cliquer sur le bouton de la barre Designer outil).
- Sélectionnez le contrôle wx.ListCtrl sur la «Liste des contrôles” volet et déposez-le sur le concepteur, ce sera automagiquement l'ajouter à la calibreuse bsMain.
- Sur la page “Conteneurs / Mise en page” sélectionner le volet wx.FlexGridSizer et déposez-le aussi sur le wx.Panel, qui encore une fois automagiquement l'ajouter à la calibreuse bsMain.
- Cliquez sur le “calibreurs” et de sélectionner le flexGridSizer1 et renommez-le par exemple fgsFields.
- Dans l'Inspecteur modifier le paramètre «Cols» de '0 'à '2' et le paramètre 'Rows' de '1 'à '0', que nous aurons à des colonnes de contrôles et widgets dans ce calibreur.
- Poster les modifications, sauvegardez le fichier et ouvrez le Concepteur de nouveau. Je ne quitte ce regurarely veiller à ce que je ne perd pas trop de mon travail si quelque chose doit aller mal. Il est aussi une bonne idée de simplement exécuter l'application pour voir à quoi il ressemble.
- Dans l'Inspecteur changer le nom de «Frame1» à «AddressEntry» et le titre de «Frame1» à «Adresse formulaire d'inscription».
- Sélectionnez le wx.ListCtrl dans le Concepteur et changer le style de wx.LC_ICON à

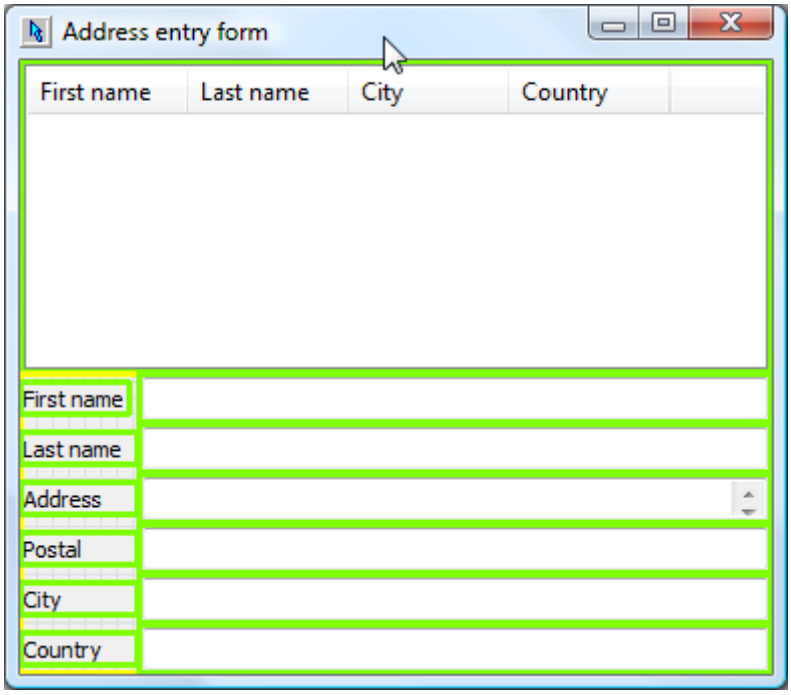
wx.LC_REPORT et sur les “accessoires”, cliquez sur le volet (colonnes), puis sur le “…” pour ouvrir l'éditeur de collection pour la listctrl. Créez les colonnes “nom, prénom, ville et pays”.

- Cliquez sur le “calibreurs” volet et double-cliquez sur bsMain pour ouvrir l'Éditeur Collection c'est. Puis cliquez sur le self.listCtrl1 et modifier la frontière de 0 à 2 (ou ce que vous trouvez approprié pour une bordure autour de ce contrôle) et de changer drapeau de 0 à wx.ALL | wx.EXPAND et changer Proportion de 0 à 1. Ces changements feront en sorte que vous disposez de 2 pixels d'espace autour de la listctrl et qu'il va utiliser autant d'espace que est disponible. Si vous exécutez cette application peu maintenant, vous verrez que le listctrl prend tout l'espace disponible.
- Sur la page “calibreurs” volet ouvrir l'éditeur de la collection pour le sizer fgsFields et ajouter 12 nouveaux produits, quand vous regardez maintenant au concepteur, il affichera ces éléments en rouge.
- Dans le volet “Commandes de base” Palette sélectionnez le contrôle wx.StaticText et déposez-le sur la partie supérieure gauche rouge et le droit de laisser tomber une wx.TextCtrl puis répétez jusqu'à ce que votre écran Concepteur et éditeur de la collection fgsFields regard quelque chose le long ces lignes.



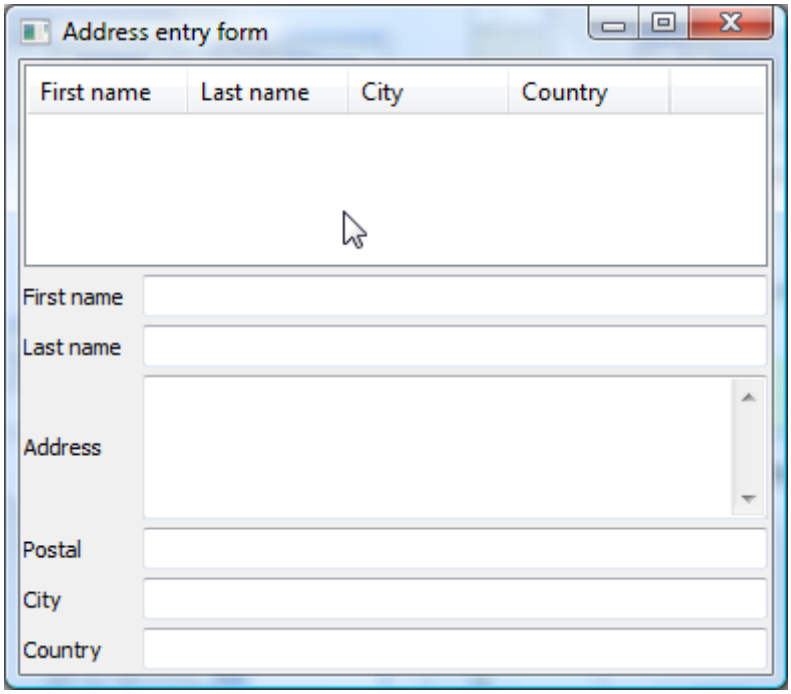
Assurez-vous de renommer les contrôles aux noms qui font sens (c.-à-nom, prénom, adresse, postalCode, ville et pays).

- Maintenant nous avons besoin pour définir la frontière, d'un drapeau et de la proportion de chacun de ces contrôles.
- Pour wx.StaticText je suggère: 2, wx.ALL | wx.ALIGN_CENTER_VERTICAL et 0
- Pour wx.TextCtrl je suggère: 2, wx.ALL | wx.EXPAND et 0
- Sur la page “calibreurs” volet, vous devez sélectionner le sizer fgsFields et de rendre le extensibles deuxième colonne que vous pouvez faire de l'inspecteur “Props” volet en cliquant sur le “…” à côté de “(Growables)”.
- Et pour que cela prenne effet, vous avez besoin de changer le drapeau de la calibreuse fgsFields dans le sizer bsMain de 0 à wx.EXPAND.
- Donc, maintenant vous devriez voir quelque chose comme ceci dans le concepteur.



Si vous l'exécutez à ce point et de redimensionner la fenêtre, vous pouvez voir les des calibreurs au travail.

- Vous remarquerez aussi que vous voyez des barres de défilement sur le champ Adresse et il est plus grand que dans d'autres domaines. Pour cela, vous avez besoin de changer son style de 0 à wx.TE_MULTILINE et dans le Concepteur de vous agrandir à la hauteur que vous souhaitez allouer pour elle.
- Lorsque vous exécutez l'application, vous devriez voir quelque chose le long de ces lignes.

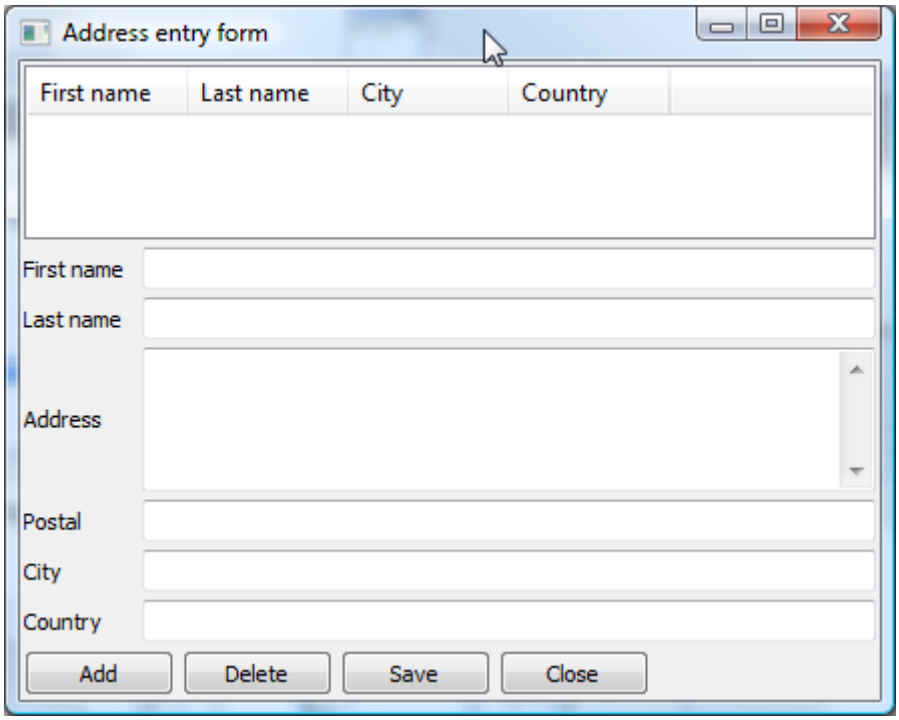


Nous aurons également besoin des boutons pour cela, donc nous pouvons ajouter, supprimer, enregistrer et fermer ce formulaire.

- Pour cette ouverture du Concepteur de nouveau et déposer une autre flexGridSizer (Je vais le

nommer fgsButtons) sur le “calibreurs” onglet, puis l'ajouter à la calibreuse bsMain.

- Puis ajouter quatre éléments à la calibreuse fgsButtons puis déplacer des contrôles sur les wx.Button carrés rouges sur le concepteur.
- Dans l'éditeur de collection sizer changer la frontière à 2, le drapeau à wx.ALL pour tous ces boutons.
- Ensuite, sélectionnez le premier bouton en double cliquant sur son entrée dans l'éditeur de collections et de l'inspecteur “cons” volet modifier l'étiquette de button1 à “” (vide) et le nom de button1 à “ajouter” et l'ID à wx.ID_ADD.
- Répétez cette opération pour les autres, mais nommer les supprimer, enregistrer et fermer et d'utiliser les entrées appropriées (wx.ID_ avoir accès à qui est nouveau dans Boa 0.6.0 l'ID du bouton de stock, il ne fonctionnera que si vous effacer l'étiquette.)



Vous devriez maintenant voir quelque chose comme ci-dessus lorsque vous l'exécutez.

De toute évidence vous avez seulement le code GUI à ce point et il faudrait à la chair tout cela avec le code pour chacun des boutons, mais pour le moment cela va au-delà de ce tutoriel.

S'il vous plaît noter que le fichier généré au cours de cet exemple est également disponible dans le répertoire “Exemples \ guide” sous votre répertoire d'installation Boa.

Pour le codage des lignes de guidage vous pouvez également consulter le guide de style wxPython http://wiki.wxpython.org/index.cgi/wxPython_Style_Guide.

1)

EDI pour Python