

Sublime Text でプラグインを作ろう!

ゆーてー

1. はじめに

Sublime Text¹というテキストエディタがあります。このエディタは、様々な人が提供しているプラグインを導入することで機能を拡張し、容易に自分好みのエディタに改造することができます。また、プラグインを自作することも可能なので、「ぼくのかんがえたさいきょうのエディタ」を実現することができます。今回は Sublime Text のプラグインの作り方を実例に沿って紹介していきます。コードは Python²で記述されるので、知識があると捗ります。

2. 準備

まずプラグイン作成の準備をしましょう。 *Tools -> New Plugin* を選択します。すると何やらソースコードのようなものが生成されます(図1)。これがプラグイン本体になります。ひとまずこの状態で動作確認をしてみたいのでファイルを保存しておきます。デフォルトでは User ディレクトリに保存されますが、今は一つ上の Packages ディレクトリに新しくディレクトリを作り、その中に保存しましょう。理由は後程説明します。

保存できたら、 *View -> Show Console* を選択し、コンソールを表示させます。コンソールに `view.run_command(' example')` というコマンドを入力し実行すると、ファイルの先頭に " Hello, World!" という文字列が挿入されます(図2)。

これでプラグインの生成と実行までができるようになりました。これを改変していき、お好みのプラグインを作っていくわけですね。

```
1 import sublime, sublime_plugin
2
3 class ExampleCommand(sublime_plugin.TextCommand):
4     def run(self, edit):
5         self.view.insert(edit, 0, "Hello, World!")
6
```

図 1. 自動生成されたコード

¹ Sublime Text - <http://www.sublimetext.com/>

² Python - <https://www.python.org/>

```
1 Hello, World!import sublime, sublime_plugin
2
```

図 2. Hello, World!

3. 命名規則

コードの 3 行目に *ExampleCommand* というものがありますが、この “ Example” の部分がコマンド名となります。どんな名前にしてもいいわけではなく、必ず最初の文字は大文字、2 単語以上繋げる場合も、各単語の頭文字は大文字にしてください。コマンド実行の際は、すべて小文字にして、単語間を “ _ ” (アンダーバー) で繋いだ文字列を渡します。

例えば、「NiceVimCommand」 という名前を付けると

```
view.run_command(' nice_vim' )
```

で実行できます。

4. 作成

では実際に作るものですが、こちら (<https://github.com/yuutee/NiceVim>) に置いてある NiceVim というものを作ってみましょう。ソースコードは以下のような感じになってます。

```
1 # -*- coding: utf-8 -*-
2 import sublime, sublime_plugin
3 emacs2 = """Emacs の アイコン だよ(ページの都合上省略)"""
4
5 class NiceVimCommand(sublime_plugin.TextCommand):
6     def run(self, edit):
7         region_all = sublime.Region(0, self.view.size())
8
9         self.view.replace(edit, region_all, emacs2)
10
11     if self.view.file_name():
12         self.view.set_scratch(True)
13         self.view.run_command("save")
```

NiceVim は編集時のファイルに Emacs のアイコンをアスキーアート変換したものを上書きするプラグインです(図 3, 4)。

必要そうな機能を書き出すと以下のようになります。

- ファイルの中身を全選択
- 選択範囲を指定した文字列で置換
- 保存

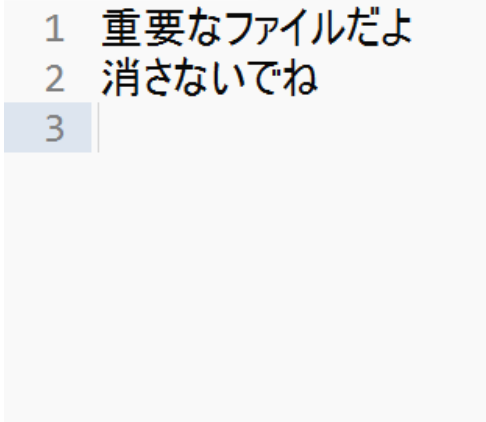


図 3. Before

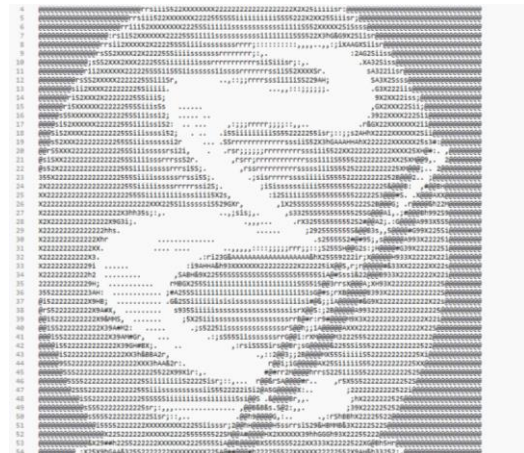


図 4. After

これらを実現できそうなクラスやメソッドを Sublime Text API Reference ³から探してみます。

まず全選択ですが、Sublime API では選択範囲を指定する Region というクラスが存在するので、これを使ってみましょう。sublime.Region(a, b)という記述で、ファイル中の a 地点から b 地点までの範囲を選択します。全選択したいのでここではファイルの先頭から末尾までを指定してあげればいいでしょう。

```
region_all = sublime.Region(0, self.view.size())
```

これで全選択完了です。

次に文字列の置換ですが、View というクラスメソッドの replace を使います。replace(edit, region, string)という記述で呼び出します。ここで書き換える必要があるのは region と string です。region には先程の region_all を、string には適当な文字列を突っ込んであげると良いでしょう。ここでは Emacs のアイコンをアスキーアート変換した文字列 emacs2 を指定します。

```
self.view.replace(edit, region_all, emacs2)
```

³ Sublime Text API Reference - https://www.sublimetext.com/docs/3/api_reference.html

最後に保存部分ですが、コンソールでコマンドを実行するのと同様に、プラグイン中でも既存のコマンドを実行することが可能です。なので、保存時に呼び出すコマンドを探してきてコード中で実行するように書いてあげましょう。実行できるコマンド一覧はこちら(<http://www.sublimetext.com/docs/commands>)。呼び出すのは save コマンドなのでこんな感じ。

```
self.view.run_command("save")
```

これで保存ができるようになりました。ここまでで必要な機能の実装ができたので実行してみましょう。きっと素敵なファイルになります。

5. おわりに

プラグイン作成の話はここまでです。これを読んだ皆さんも独自の嫌がらせプラグインを作ったり、NiceVim を拡張したりして遊んでみてください。ここまで読んでいただきありがとうございました。