Python を眺めてみる

yassu

2015/07/25 (Sat)

お前だれよ

- 名前: yassu[0320]
- 職業: 某大学院数学科 M2 (専攻: 特異点論)
- Python 歴: 3 年くらい
- Vim 使い

PyCon mini @ sapporo

Agenda

- Python とは?
- Python の利点と欠点
- Python のライブラリ
- Python が使われているプロジェクト
- PEPS
- Python の禅

Python とは?

- 凡庸プログラミング言語
- ドイツ人の Guido van Rossum によって作られた
- モンティ・パイソンというアニメから言語名は命名された
- ヘビのアイコンが目印
- Ruby のお兄さん的存在

Python の利点と欠点

- コードのリーダビリティが高くなるように言語が設計されていると主張されている
- 型付けが弱い
- (C に比べると) 遅い
- Python の標準的なコーディング規約なる Pep8 もの存在する.
- “出来るだけ同じコードとして書けるなら、同じように書いた方がいい”

Python の利点と欠点 (2)

- カプセル化の概念がない
- インターフェースの概念が後付け (duck typing, “それがアヒルのように歩き、アヒルのように鳴くのなら、それはアヒルである”)
- python 2.x から 3.x への移行で少々の変更が加えられた

コードを眺めてみる

初めての Python プログラム (dive into python3)

Python のライブラリ

- numpy: 数値計算モジュール
- matploblib: グラフ作成ライブラリ
- sympy: 科学計算, 数値計算ライブラリ (pure python)
- pandas: データ解析のライブラリ
- scipy: 科学計算・数値計算ライブラリ
- Django: Web フレームワーク
- Blender: 3D, アニメーション, ゲーム制作環境
- PILLOW: 画像処理ライブラリ

Python が使われているプロジェクト

- reStructuredText (RST): グラフィカルな markdown みたいなの
- Sphinx (およびそのプラグイン): 構造的でもっとすごい rst みたいなの
- SCons: Python で書ける Makefile 的なもの
- Trac
- Sage: 数式処理システム. python のライブラリでもある.
- Impressive
- blockdiag (およびそのプラグイン): シンプルなテキストからブロック図を生成する ブロック図生成ツール
- vint: vim の lint

What is PEPS ?

- 以後 Ref: “パーフェクト Python”
- Python の設計のプロセスを可視化していき, 実装の前に皆の意見を得ることを目的とする

重要なところ

- PEP 0 (Indenx of Python Enhancement Proposals): PEP の目次
- PEP 1 (PEP Purpose and Guidelines): PEP についてのガイドライン
- PEP 5 (Guidelines for Language Evolution): 後方互換精を崩す際の決め事
- PEP 8 (Style Guide for Python Code): 標準のコーディング規約

Python の禅

```
>>> import this
```

Beautigul is better than ugly. (醜悪より美しいほうが良い)

- コードを “美しく” 保つ “ことは, 少々開発が遅くなることより大事.

Explicit is better than implicit. (暗黙より明示するほうが良い)

例えば

```
>>> from re import *
```

よりも

```
>>> from re import search, compile
```

Python では暗黙的な変数の使用も許されない.

ただし, ipython での_のように, 例外もある.

Flat is better than nested . (ネストしたものよりフラットなほうが良い)

例えばモジュール名は xxx.yyy.zzz ではなく xxx_yyy_zzz などとする.

```
if A:
    if B:
        xxx
    elif C:
        yyy
```

よりも

```
if A and B:
    xxx
elif A and C:
    yyy
```

In the face of ambiguity, refuse the temptation to guess. (曖昧なモノに出くわしたら推測してはいけない)

$1 +$ ”0” は 1 ではない.

There should be one- and preferably only one -obvious way to do it. (1つのことをするのに, いろいろなやり方は好ましくない)

- いろいろな人がみんな知らない特殊なプログラムの書き方をしている状況を想像してみよう

if the implementation is easy to explain, it's good idea. (実装の説明が簡単? そのアイデアは良いのでしょう)

程よくシンプルに書けているというのはいいこと.

Namespaces are one honking gread idea - let's do more of thosse！(ネームスペースはすごく良いアイデアの1つ. もっと考えよう)

ネームスペースによって値のコンフリクト (衝突) のミスが少ないプログラムが書けるようになるだろう.

ありがとうございました.