寻找自我的博客

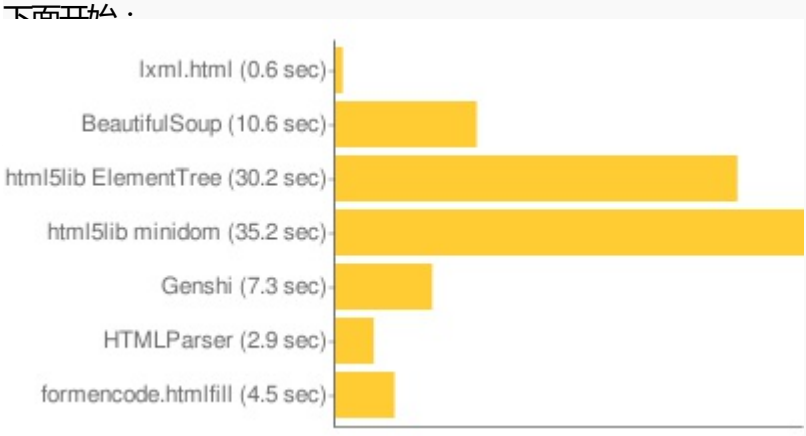
Python 开发者应该知道的 7 个开发库

分类： Python 2012-11-18 19:16 72人阅读 评论(0) 收藏 举报

在我多年的 Python 编程经历以及在 Github 上的探索漫游过程中，我发掘到一些很不错的 Python 开发包，这些包大大简化了开发过程，而本文就是为了向大家推荐这些开发包。

请注意我特别排除了像 SQLAlchemy 和 Flask 这样的库，因为其实在太优秀了，无需多提。

下面开始：



这个图里我们发现 lxml 的性能是如此之好，不过文档就很少，而且使用上相当的笨拙！那么是选择一个使用简单但是速度奇慢的库呢，还是选择一个速度飞快但是用起来巨复杂的库呢？

谁说二者一定要选其一呢，我们要的是用起来方便，速度也一样飞快的 XML/HTML 解析库！

而 PyQuery 就可以同时满足你的易用性和解析速度方面的苛刻要求。

看看下面这几行代码：

```
from pyquery import PyQuery
page = PyQuery(some_html)

last_red_anchor = page('#container > a.red:last') 很简单吧，很像是 jQuery，但它却是 Pyth
```

不过也有一些不足，在使用迭代时需要对文本进行重新封装：

```
for paragraph in page('#container > p'):
    paragraph = PyQuery(paragraph)
    text = paragraph.text()
```

2. dateutil

安装方法：pip install dateutil

处理日期很痛苦，多亏有了 dateutil

```
from dateutil.parser import parse

>>> parse('Mon, 11 Jul 2011 10:01:56 +0200 (CEST)')
datetime.datetime(2011, 7, 11, 10, 1, 56, tzinfo=tzlocal())

# fuzzy ignores unknown tokens

>>> s = """Today is 25 of September of 2003, exactly
...         at 10:49:41 with timezone -03:00."""
>>> parse(s, fuzzy=True)
datetime.datetime(2003, 9, 25, 10, 49, 41,
                  tzinfo=tzoffset(None, -10800))
```

3. fuzzywuzzy

安装方法：pip install fuzzywuzzy

fuzzywuzzy 可以让你对两个字符串进行模糊比较，当你需要处理一些人类产生的数据时，这非常有用。下面代码使用Levenshtein 距离比较方法来匹配用户输入数组和可能的选择。

```
from Levenshtein import distance

countries = ['Canada', 'Antarctica', 'Togo', ...]

def choose_least_distant(element, choices):
    'Return the one element of choices that is most similar to element'
    return min(choices, key=lambda s: distance(element, s))

user_input = 'canaderp'
choose_least_distant(user_input, countries)
>>> 'Canada'
```

这已经不错了，但还可以做的更好：

```
from fuzzywuzzy import process

process.extractOne("canaderp", countries)
>>> ("Canada", 97)
```

4. watchdog

安装方法：pip install watchdog

watchdog 是一个用来监控文件系统事件的 Python API和shell实用工具。

5. sh

安装方法：pip install sh

sh 可让你调用任意程序，就好象是一个函数一般：

```
from sh import git, ls, wc

# checkout master branch
git(checkout="master")

# print(the contents of this directory
print(ls("-l"))

# get the longest line of this file
longest_line = wc(__file__, "-L")
```

6. pattern

安装方法：pip install pattern

Pattern 是 Python 的一个 Web 数据挖掘模块。可用于数据挖掘、自然语言处理、机器学习和网络分析。

7. path.py

安装方法：pip install path.py

当我开始学习 Python 时，os.path 是我最不喜欢的 stdlib 的一部分。尽管在一个目录下创建一组文件很简单。

```
import os

some_dir = '/some_dir'
files = []

for f in os.listdir(some_dir):
    files.append(os.path.joinpath(some_dir, f))
```

但 listdir 在 os 而不是 os.path 中。

而有了 path.py ，处理文件路径变得简单：

```
from path import path

some_dir = path('/some_dir')

files = some_dir.files()
```

其他的用法：

```
>>> path('/').owner
'root'

>>> path('a/b/c').splitall()
[path(''), 'a', 'b', 'c']

# overriding __div__
>>> path('a') / 'b' / 'c'
path('a/b/c')

>>> path('ab/c').relpathto('ab/d/f')
path('../d/f')
```

是不是要好很多？

http://www.oschina.net/question/12_78983?from=20121118