

## ประมวลผล XML ด้วย PyXML

- 
- อ่าน 7356 ครั้ง

Python เป็นภาษาหนึ่ง ที่เป็น interactive script ซึ่งสามารถใช้งานได้หลากหลาย รวมทั้งการประมวลผลไฟล์ XML ด้วย ซึ่งสามารถทำได้ง่ายด้วย โดยใช้ module PyXML

### 1. ติดตั้ง PyXML

อันดับแรกเราต้องติดตั้ง Python Interpreter ก่อนครับ สามารถดาวน์โหลดได้จาก <http://www.python.org/> [1] ซึ่งผมขอข้ามละกัน เมื่อติดตั้งแล้ว สำหรับ Windows อาจต้อง set path ให้เรียก python interpreter ใช้งานได้ ส่วน linux ใช้ apt-get install ก็ใช้งานได้แล้วครับ อันดับต่อมาคือโหลด PyXML ซึ่งเป็น module ที่ใช้ประมวลผล XML ครับ โหลด source ได้ที่ <http://sourceforge.net/projects/pyxml/> [2] จากนั้น unzip แล้วติดตั้ง โดยเรียกคำสั่ง

```
# python setup.py install
```

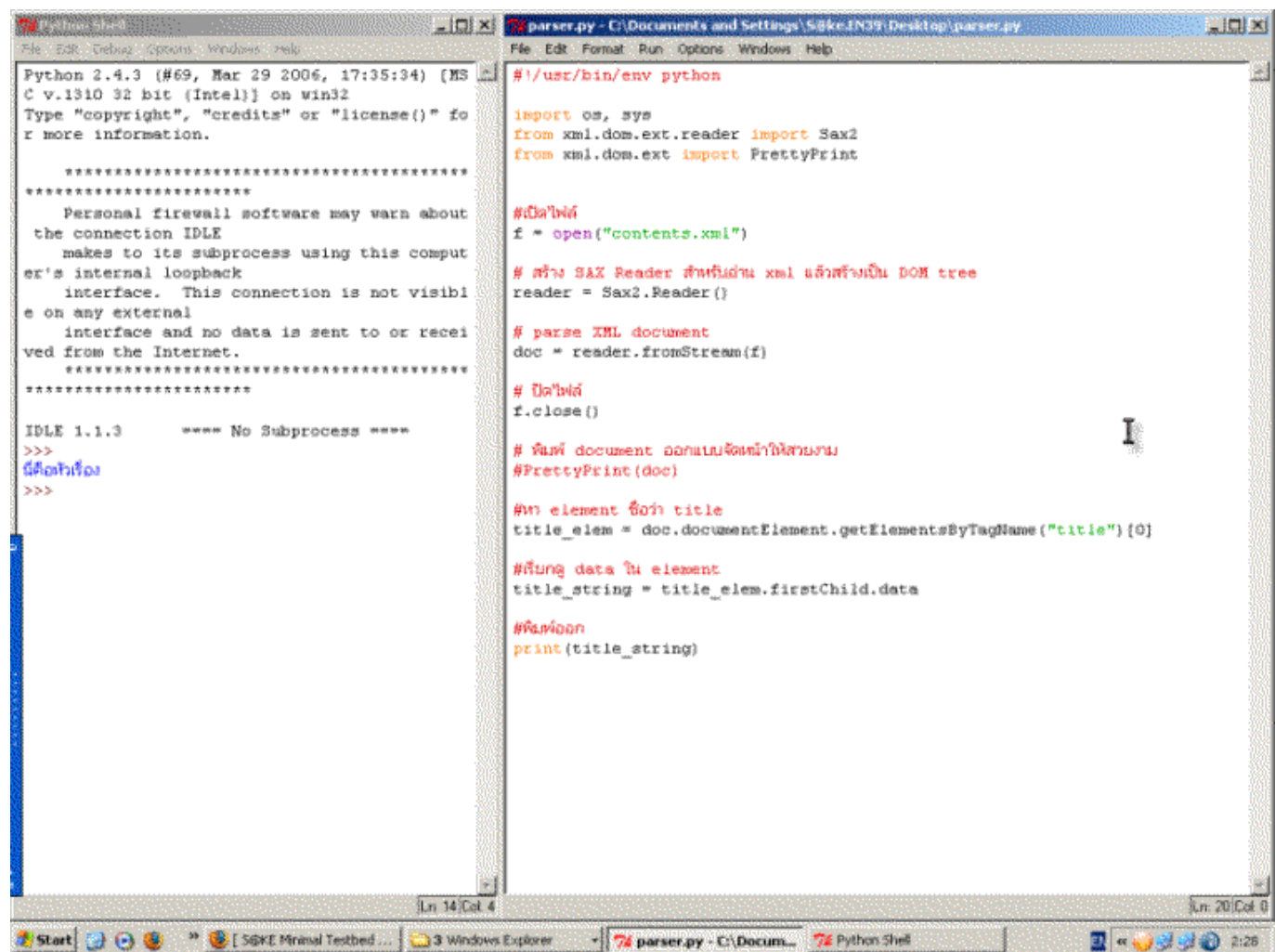
จากนั้นระบบจะทำการ install ให้ ซึ่งสำหรับ window บางครั้งอาจต้องติดตั้ง .NET SDK เพื่อ compile module ด้วย ใช้แบบ binary จะสะดวกกว่า สามารถทดสอบได้โดยเข้าไปที่ interpreter ดังนี้

```
>>> import xml
>>> xml.__version__
'0.8.4'
```

ถือว่าใช้ได้ครับ

### 2. ลงมือเขียน parser

เมื่อลง python จะมี IDLE ซึ่งเป็น IDE ที่ติดตั้งมาให้เหมาะสำหรับการทำงานพอสมควร เราจะใช้ IDE ตัวนี้เป็นหลักครับ



อันดับแรกสร้าง xml ไฟล์ที่จะตัดก่อน สมมติชื่อ contents.xml

1. `<?xml version="1.0" encoding="tis-620"?>`
2. `<contents>`
3. `<title>นี่คือหัวเรื่อง</title>`
4. `<details>นี่คือรายละเอียดของ ไฟล์ xml</details>`
5. `</contents>`

สำหรับที่จะทดสอบเขียน เราจะทดสอบโดยใช้ DOM และ SAX ครั้บ

เลือก File >> New Window

จากนั้น save file เป็น .py

1. `#!/usr/bin/env python`

```
2.     import os, sys

3.     from xml.dom.ext.reader import Sax2

4.     #เปิดไฟล์

5.     f = open("contents.xml")

6.     # สร้าง SAX Reader สำหรับอ่าน xml แล้วสร้างเป็น DOM tree

7.     reader = Sax2.Reader()

8.     # parse XML document

9.     doc = reader.fromStream(f)

10.    # ปิดไฟล์

11.    f.close()

12.    # พิมพ์ document ออกแบบจัดหน้าให้สวยงาม

13.    PrettyPrint(doc)
```

จากนั้นเลือก F5 สำหรับ รัน module

จริง ๆ แล้ว เราสามารถพิมพ์คำสั่งที่ละบรรทัดที่ interpreter เลยก็ได้ครับ แต่ผมชอบเขียนเป็นไฟล์มากกว่า เพราะแก้ไขสะดวก ผลลัพธ์จะได้เอกสาร XML ข้างต้นออกมา

หากต้องการหา element ก็สามารถใช้คำสั่ง

```
1.     #หา element ชื่อว่า title

2.     title_elem = doc.documentElement.getElementsByTagName("title")[0]

3.     #เรียกดู data ใน element

4.     title_string = title_elem.firstChild.data

5.     #พิมพ์ออก

6.     print(title_string)
```

ยังมีที่น่าสนใจอีกมากมาย สำหรับวันนี้ขอพอก่อนนะครับ

::ข้อมูลเพิ่มเติม ::

<http://pyxml.sourceforge.net/topics/howto/xml-howto.html> [3]

<http://www.xml.com/pub/a/2002/09/25/py.html> [4]

SiteTags: [programming](#) [5]

[Python](#) [6]

[XML](#) [7]

Source URL (modified on 2008-07-14 17:08): <https://sake.in.th/node/4>

### Links

[1] <http://www.python.org/>

[2] <http://sourceforge.net/projects/pyxml/>

[3] <http://pyxml.sourceforge.net/topics/howto/xml-howto.html>

[4] <http://www.xml.com/pub/a/2002/09/25/py.html>

[5] <https://sake.in.th/tags/sitetags/programming>

[6] <https://sake.in.th/category/sitetags/python>

[7] <https://sake.in.th/category/sitetags/xml>