**Можливості пакету xml**

Python інтерфейс для обробки XML зосереджується в пакеті xml, який в собі містить багато підпакетів, серед яких:

1. **xml.etree.ElementTree**: простий і легкий у керуванні XML процесор
2. **xml.dom**: DOM API
3. **xml.dom.minidom**: мінімальна реалізація DOM
4. **xml.dom.pulldom**: підтримка для побудованих часткових DOM дерев
5. **xml.sax**: базові SAX2 класи та деякі функції для зручної роботи
6. **xml.parsers.expat**: the Expat parser binding

Розберемо пакет **xml.dom.minidom**, так як він дозволяє виконувати всі необхідні базові операції із xml.

Доступ до нього здійснюється через пакет xml.

**import xml.dom.minidom**

Або інший варіант:

**from xml.dom import minidom**

Парсер підтримує завантаження XML документів як із файлу, так і звичайної рядка. Об'єкт парсеру створюється в залежності від джерела XML даних (файл або рядок) одним з наступних методів.

1. Завантаження з файлу:

**dom = xml.dom.minidom.parse("filename.xml")**

2. Завантаження з рядка:

**dom = xml.dom.minidom.parseString(xmlStr)**

По завершенню ініціалізації рекомендується викликати метод normalize для злиття різних фрагментів тексту в одне ціле. В іншому випадку при роботі із XML можуть виникнути помилки.

**dom.normalize()**

Уявимо, що ми підключили цей пакет, завантажили xml із файлу чи рядка і далі хочемо опрацювати ці дані.

Для пошуку вузлів за назвою тега є метод **getElementsByTagName(tagName)**. Цей метод повертає список тегів з іменем, яке ми вказали як параметр. Якщо тег з таким іменем лише один, то можна відразу звернутися до нього за індексом 0.

**node1 = dom.getElementsByTagName ( "node1") [0]**

Назву тегу можна отримати за допомогою властивості **nodeName**

**print( "name =" + node1.nodeName)**

Значення атрибута за його іменем можна отримати за допомогою методу **getAttribute(attributeName)**.

**print("attr =" + node1.getAttribute ( "attr"))**

Також значення атрибута тегу можна отримати за допомогою attributes, звернувшись до нього за індексом.

**print("attr =" + node1.attributes.item (0) .value)**

Отримати значення тега трохи складніше. Для цього необхідно спочатку звернутися до першого дочірнього вузла через колекцію childNodes і після цього отримати саме значення через властивість nodeValue.

**print("value =" + node1.childNodes [0] .nodeValue)**

Також можна створити xml документ за допомогою виклику методу об'єкта “DOM Implementation”. Ви можете отримати цей об'єкт або шляхом виклику функції **getDOMImplementation()** в пакеті **xml.dom** або модуля **xml.dom.minidom**. Якщо у вас є документ, то ви можете додати дочірні вузли до них, щоб заповнити DOM наступним чином:

**from xml.dom.minidom import getDOMImplementation  
  
 impl = getDOMImplementation()  
  
 newdoc = impl.createDocument(None, "some\_tag", None)  
 top\_element = newdoc.documentElement  
 text = newdoc.createTextNode('Some textual content.')  
 top\_element.appendChild(text)**