**Можливості XML**

Розширювана мова розмітки (англ. Extensible Markup Language, скорочено XML) — запропонований консорціумом World Wide Web стандарт побудови мов розмітки ієрархічно структурованих даних для обміну між різними застосунками, зокрема, через Інтернет. XML-документ складається із текстових знаків, і придатний до читання як комп’ютером, так і людиною.

Цілі дизайну XML є підкреслення простоти, універсальність і зручність використання через Інтернет. Це текстовий формат даних з сильною підтримкою через Unicode для різних людських мов. Хоча дизайн XML фокусується на документах, мова широко використовується для представлення довільних структур даних, наприклад, таких, що використовуються в веб-службах.

Кілька систем схеми існують, щоб допомогти у визначенні мов, базованих на XML, в той час як програмісти розробили безліч інтерфейсів прикладного програмування (API) для полегшення обробки даних XML.

XML є надзвичайно корисним для відстеження даних відносно малого або середнього обсягу без необхідності використання «хребта», зробленого на основі SQL.

Стандартна бібліотека Python забезпечує мінімальний, але корисний набір інтерфейсів для роботи з XML.

Двома найбільш основними і широко використовуваними API для даних XML є SAX і DOM інтерфейси:

* Простий API для XML (SAX - Simple API for XML): тут вводяться зворотні виклики для подій, що представляють інтерес, а тоді парсер «проходиться» документом. Це корисно, коли документи великі або є обмеження пам'яті, так як він розбирає файл, зчитуючи його з диска, і весь файл ніколи не зберігається в пам'яті.
* Document Object Model (DOM) API: Це рекомендація World Wide Web Consortium, в якому весь файл зчитується в пам'ять і зберігається у вигляді ієрархічної (деревовидної) форми, щоб представити всі особливості документа XML.

SAX, очевидно, не може обробляти інформацію так швидко, як DOM може при роботі з великими файлами. З іншого боку, використовування виключно DOM дійсно може вбити ресурси, особливо, якщо він використовується для роботи з багатьма маленькими файлами.

SAX тільки для читання, в той час як DOM дозволяє вносити зміни в файл XML. Так як ці два різних API в буквальному сенсі доповнюють один одного, немає ніяких причин, чому не можна їх обох використовувати одночасно для великих проектів.

Python інтерфейси для обробки XML групуються в пакеті xml.

Важливо відзначити, що модулі в пакеті XML вимагають, щоб там був доступний принаймні один SAX-сумісний XML парсер. Expat парсер уже входить в Python, тому модуль xml.parsers.expat завжди буде доступний.

Є такі підмодулі обробки XML:

* [xml.etree.ElementTree](https://docs.python.org/3.5/library/xml.etree.elementtree.html#module-xml.etree.ElementTree): the ElementTree API, простий і легкий процесор XML, що забезпечує аналіз і створення даних XML. XML є за своєю суттю ієрархічний формат даних, і тому найкраще представляється деревом. ET має два класи для цієї мети - ElementTree представляє весь документ XML у вигляді дерева, а Element являє собою окремий вузол в цьому дереві. Взаємодія з усім документом (читання і запис в / з файлів), як правило, робиться на рівні ElementTree. Взаємодія з одного елемента XML і його вкладені елементи виконуються на рівні Element.
* [xml.dom](https://docs.python.org/3.5/library/xml.dom.html#module-xml.dom): the DOM API definition, об'єктна модель документа, або «DOM» є міжмовним API від World Wide Web Consortium (W3C) для доступу і модифікації XML-документів. Реалізація DOM презентує документ XML як структуру у вигляді дерева, або дозволяє створення такої структури з нуля. Потім ця модель дає доступ до структури через набір об'єктів, що надають добре відомі інтерфейси.
* [xml.dom.minidom](https://docs.python.org/3.5/library/xml.dom.minidom.html#module-xml.dom.minidom): мінімальна реалізація інтерфейсу об'єктної моделі документа, з API, аналогічним на інших мовах. Він призначений, щоб бути простішим і значно меншим, ніж повний DOM. DOM програми зазвичай починають з парсингу деяких XML в DOM. З xml.dom.minidom, це робиться за допомогою функції синтаксичного аналізу.
* [xml.dom.pulldom](https://docs.python.org/3.5/library/xml.dom.pulldom.html#module-xml.dom.pulldom): підтримка для створення часткових DOM дерев. Модуль xml.dom.pulldom забезпечує спеціальний парсер, який за необхідності може зробити фрагменти документа з доступом до DOM. Основна концепція включає в себе витяг «подій» з потоку вхідного XML і їх обробкою.
* [xml.sax](https://docs.python.org/3.5/library/xml.sax.html#module-xml.sax): SAX2 базові класи і функції зручності. Пакет xml.sax надає ряд модулів, що реалізують Simple API для інтерфейсу XML (SAX) для Python. Сам пакет надає виключення SAX і допоміжні функції, які будуть найбільш використовувані користувачами SAX API.
* [xml.parsers.expat](https://docs.python.org/3.5/library/pyexpat.html#module-xml.parsers.expat): Expat парсер зв’язування. Модуль xml.parsers.expat є інтерфейсом Python для Expat XML парсера. Модуль забезпечує єдиний тип розширення, xmlparser, який представляє поточний стан XML-парсеру. Після того як об'єкт xmlparser був створений, різні атрибути об'єкта можуть бути встановлені в функції обробника. Коли XML документ потім подається в парсер, функції обробника викликаються для даних символів і розмітки в документі XML. Цей модуль використовує pyexpat модуль для забезпечення доступу до парсера Expat.

Також для парсингу XML можна використовувати і інші бібліотеки:

* [untangle](https://github.com/stchris/untangle) - проста бібліотека, яка приймає документ XML і повертає об'єкт Python, який відображає вузли та атрибути в його структурі
* [xmltodict](https://github.com/martinblech/xmltodict) - ще одна проста бібліотека, яка спрямована на те, аби робота з XML відчувалась як робота з JSON