





(/page-cours-python-debutant-  
documentation)

(/page-cours-python-avance-  
documentation-francaise)

 (/page-django-cours-tutoriels-  
documentation-francaise)

 (/page-raspberry-pi-apprendre-debutant-  
acheter-python)



## Un document XML

Voici un exemple de document XML, que nous avons enregistré sous le nom de data.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<users>
  <user data-id="101">
    <nom>Zorro</nom>
    <metier>Danseur</metier>
  </user>
  <user data-id="102">
    <nom>Hulk</nom>
    <metier>Footballeur</metier>
  </user>
  <user data-id="103">
    <nom>Zidane</nom>
    <metier>Star</metier>
  </user>
  <user data-id="104">
    <nom>Beans</nom>
    <metier>Epicier</metier>
  </user>
  <user data-id="105">
    <nom>Batman</nom>
    <metier>Veterinaire</metier>
  </user>
  <user data-id="106">
    <nom>Spiderman</nom>
    <metier>Veterinaire</metier>
  </user>
</users>
```

La première ligne indique l'encodage, on reste toujours dans l'encodage UTF-8. Ensuite on remarque que la balise "users" possède d'autres balises "user" qui elles même possèdent leurs propres balises. Les données sont hiérarchisées dans un arbre et chaque noeud apporte une information.

## Lire un XML

Voici un petit script qui affiche tous les noms des [user](#).

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

from lxml import etree

tree = etree.parse("data.xml")
for user in tree.xpath("/users/user/nom"):
    print(user.text)
```

La liste suivante devrait s'afficher:

Zorro  
Hulk  
Zidane  
Beans  
Batman  
Spiderman

Vous pouvez afficher les [attributs](#) des balises:

```
tree = etree.parse("data.xml")
for user in tree.xpath("/users/user"):
    print(user.get("data-id"))
```

Résultat:

101  
102  
103  
104  
105  
106

Vous pouvez affiner l'affichage en proposant de n'afficher que les [user](#) dont le métier est [Veterinaire](#).

```
tree = etree.parse("data.xml")
for user in tree.xpath("/users/user[metier='Veterinaire']/nom"):
    print(user.text)
```

Ce qui devrait afficher:

Batman  
Spiderman

## Créer un xml

Vous pouvez créer un XML:

```
users = etree.Element("users")
user = etree.SubElement(users, "user")
user.set("data-id", "101")
nom = etree.SubElement(user, "nom")
nom.text = "Zorro"
metier = etree.SubElement(user, "metier")
metier.text = "Danseur"
print(etree.tostring(users, pretty_print=True))
```

## Résultat

```
<users>
  <user data-id="101">
    <nom>Olivier</nom>
    <metier>Danseur</metier>
  </user>
</users>
```

## Code pour arriver au document XML initial:

```
users = etree.Element("users")

users_data = [
    ("101", "Zorro", "Danseur"),
    ("102", "Hulk", "Footballeur"),
    ("103", "Zidane", "Star"),
    ("104", "Beans", "Epicier"),
    ("105", "Batman", "Veterinaire"),
    ("106", "Spiderman", "Veterinaire"),
]


for user_data in users_data:
    user = etree.SubElement(users, "user")
    user.set("data-id", user_data[0])
    nom = etree.SubElement(user, "nom")
    nom.text = user_data[1]
    metier = etree.SubElement(user, "metier")
    metier.text = user_data[2]

print(etree.tostring(users, pretty_print=True))
```

## Les méthodes des noeuds

Si vous voulez approfondir vos connaissances sur la lib [lxml](#), il existe plein d'autres méthodes associées aux noeuds que vous pouvez voir en lançant la commande `help(<noeuds>)`. L'aide est en anglais et comme je suis sympa, je vous l'ai traduite:

<code>addnext(element)</code>	: Ajoute un élément en tant que frère juste après l'élément
<code>addprevious(element)</code>	: Ajoute un élément en tant que frère juste avant l'élément
<code>append(element)</code>	: Ajoute un sous-élément à la fin de l'élément
<code>clear()</code>	: Supprime tous les sous-éléments
<code>extends(elements)</code>	: Etend les éléments passé en paramètre
<code>find(path)</code>	: Recherche le premier sous-élément qui correspond au tag/path
<code>findall(path)</code>	: Recherche tous les sous-éléments qui correspondent au tag/path
<code>findtext(path)</code>	: Trouve le texte du premier sous-élément qui correspond au tag/path
<code>get(key)</code>	: Recupère l'attribut d'un élément
<code>getchildren()</code>	: Retourne tous les enfants d'un élément ! attention déprécié pour <i>List</i>
<code>getnext()</code>	: Retourne le prochain élément frère
<code>getparent()</code>	: Retourne le parent de l'élément
<code>getprevious()</code>	: Retourne l'élément frère précédant
<code>index(child)</code>	: Trouve la position de l'élément
<code>insert(index)</code>	: Insère un sous-élément à la position indiquée
<code>items()</code>	: Retourne les attributs d'un élément (dans un ordre aléatoire)
<code>keys()</code>	: Retourne une liste des noms des attributs
<code>remove(element)</code>	: Supprime l'élément passé en paramètre
<code>replace(el1, el2)</code>	: Remplace el1 par el2
<code>set(key, value)</code>	: Créer un attribut avec une valeur
<code>values()</code>	: Retourne les valeurs des attributs
<code>xpath(path)</code>	: Evalue une expression xpath

Pour des informations complémentaires je vous conseille ce site: [lxml.fr](http://lxml.fr)   
(<http://lxml.de/tutorial.html>).

◀ Programmation asynchrone (/page-programmation-asynchrone-python-thread-threading)

BeautifulSoup / HTML parser ▶ (/page-beautifulsoup-html-parser-python-library-xml)

## Python débutant

Présentation Python (/)
Installer Python (/page-apprendre-installer-python-ordinateur)
Interpréteur python (/page-utiliser-interpreteur-python)
Editeurs python (/page-editeurs-python-gratuits-payants-ide)
Calculs et variables (/page-apprendre-variables-debutant-python)
Les listes (/page-apprendre-listes-list-tableaux-tableaux-liste-array-python-cours-debutant)
Les tuples (/page-apprendre-tuples-tuple-python)
Les dictionnaires (/page-apprendre-dictionnaire-python)

<a href="#">Les fonctions (/page-apprendre-creer-fonction-en-python)</a>
<a href="#">Les fonctions prédéfinies (/page-builtin-built-in-fonctions-internees-python)</a>
<a href="#">Conditions if elif else (/page-apprendre-conditions-structures-conditionnelles-if-else-python-cours-debutant)</a>
<a href="#">Boucle for / while (/page-apprendre-boucles-python-loop)</a>
<a href="#">Les modules/packages (/page-python-modules-package-module-cours-debutants-informatique-programmation)</a>
<a href="#">Les exceptions (/page-apprendre-exceptions-except-python-cours-debutant)</a>
<a href="#">Les compréhensions de liste (/page-comprehension-list-listes-python-cours-debutants)</a>
<a href="#">Programmation orientée objet (/page-apprendre-programmation-orientee-objet-poo-classes-python-cours-debutants)</a>
<a href="#">Les décorateurs (/page-decorateurs-decorator-python-cours-debutants)</a>
<a href="#">Les itérateurs/générateurs (/page-iterateurs-iterator-generateur-generator-python)</a>
<a href="#">Expressions régulières (/page-expressions-regulieres-regular-python)</a>
<a href="#">Lire / Editer un fichier (/page-lire-ecrire-creer-fichier-python)</a>

## Python avancé

<a href="#">Différences python 2 / 3 (/page-syntaxe-differente-python2-python3-python-differences)</a>
<a href="#">Encodage python (/page-python-encodage-encode-decode-unicode-ascii-codec-character-accents-probleme-string-utf8)</a>
<a href="#">Pip installe vos librairies (/page-pip-installer-librairies-automatiquement)</a>

<a href="#">Virtualenv (/page-virtualenv-python-environnement-virtuel)</a>
<a href="#">Debug (/page-debogueur-debugger-debug-python-ipython-pdb-pdbpp-cours-tutoriel-script)</a>
<a href="#">Python Path (/page-python-path-pythonpath)</a>
<a href="#">Gestion fichiers et dossiers (/page-gestion-fichiers-dossiers-python)</a>
<a href="#">Interface graphique tkinter (/page-tkinter-interface-graphique-python-tutoriel)</a>
<a href="#">Graphiques scientifiques (/page-creer-graphiques-scientifiques-python-apprendre)</a>
<a href="#">Programmation asynchrone (/page-programmation-asynchrone-python-thread-threading)</a>
<a href="#">XML et python (/page-xml-python-xpath)</a>
<a href="#">BeautifulSoup / HTML parser (/page-beautifulsoup-html-parser-python-library-xml)</a>
<a href="#">Créer un exécutable (/page-cx_freeze-creer-executables-python-cours-apprendre)</a>
<a href="#">Dossier partagé / samba (/page-dossier-partage-samba-reseau-python-fstab)</a>
<a href="#">FTP (/page-python-ftp-creer-copier-voir-dossier-folder-fichiers)</a>
<a href="#">Fabric SSH (/page-python-ssh-connect-host-distant-sftp-os-fabric-client)</a>
<a href="#">Envoyer un mail SMTP (/page-python-envoyer-mail-smtp)</a>
<a href="#">Réseau / socket (/page-reseaux-sockets-python-port)</a>
<a href="#">Base de données (/page-database-data-base-donnees-query-sql-mysql-postgre-sqlite)</a>
<a href="#">Créer un serveur web (/page-python-serveur-web-creer-rapidement)</a>
<a href="#">Websocket &amp; Crossbar (/page-websocket-python-crossbar-</a>

[push-socket-publish-subscribe-call-remote-register-wamp](#))

## Django

[Présentation Django \(/page-django-introduction-python\)](#)

[Installer Django \(/page-django-installer-python\)](#)

[Initialiser un projet Django \(/page-django-initialiser-creer-python\)](#)

[Créer une application Django \(/page-django-application-creer-python\)](#)

[ORM Django \(/page-django-orm-apprendre-base-donnees-database-queryset-modeles\)](#)

[Login Django \(/page-django-login-authentication-de-base-apprendre-creer-cours-tutoriel-python\)](#)

[Les champs des modèles \(/page-django-champs-modeles-orm-model\)](#)

[Interface admin Django \(/page-django-interface-admin-administration-settings-django-contrib-auth\)](#)

[Queryset \(/page-django-query-set-queryset-manager\)](#)

[Many to many relation \(/page-django-many-to-many-manytomany-relation-relationships-orm\)](#)

[Les vues \(/page-django-vues-views-generic-generiques-python\)](#)

[Les formulaires \(/page-django-formulaires-forms-champs-fields\)](#)

[CSRF Token \(/page-django-csrf-token-securite-formulaires-python-ajax-post-jquery-angularjs\)](#)

[Les middlewares \(/page-django-middlewares-vues-views-request-response-process-before-after-avant-apres\)](#)

[Templates Django \(/page-django-templates-structure-template-HTML-variables-boucles-tags-filtre-page-dry-application-projet-modifier-dossier\)](#)



<a href="#">Context Processor (/page-django-context-processor-template)</a>
<a href="#">Signaux Django (/page-django-signaux-signal-signals-python-cours-apprendre-tutoriel-exemple)</a>
<a href="#">Xadmin (/page-django-xadmin-bootstrap-twitter-theme-export-xls-csv-xml-python-web-plugin)</a>
<a href="#">Django select2 (/page-django-admin-select2-admin-ajax-many-to-many-foreign-key-form-widget-django_select2)</a>
<a href="#">Crispy Forms (/page-django-crispy-form-formulaire-bootstrap-twitter-html-fieldset-inline-dynamique-template)</a>
<a href="#">AngularJS et Django (/page-django-angularjs-angular-js-coffee-script-coffeescript-python-cours-tuto-tutoriel-exemples)</a>
<a href="#">Uploader un fichier en ajax (/page-django-comment-uploader-fichier-image-cours-apprendre-python-upload-file)</a>
<a href="#">Django Rest Framework (/page-django-rest-framework-drf-cours-tuto-tutoriel-exemples)</a>
<a href="#">Django déploiement (/page-django-deploiement-python-mise-en-production-linux-gunicorn-supervisor-nginx-installer-configurer-serveur-http-web)</a>

## Raspberry pi

<a href="#">Présentation Raspberry pi (/page-raspberry-pi-presentation-python-c-est-quoi-apprendre-nano-ordinateur-video)</a>
<a href="#">Installer Raspbian (/page-raspberry-pi-install-installer-debian-linux-windows-ubuntu)</a>
<a href="#">IP statique (/page-raspberry-pi-adresse-ip-statique-address-static)</a>
<a href="#">Samba / partager un dossier (/page-raspberry-pi-partage-share-dossier-folder-samba-debian)</a>
<a href="#">Lire vidéo 1080p (/page-raspberry-pi-lecture-lire-video-1080p-full-hd-omxplayer-ligne-commande)</a>

Installer VPN (/page-raspberry-pi-installer-vpn-dns-openvpn-debian-python)

Client torrent (/page-raspberry-pi-installer-client-torrent-transmission-download-dl-python)

Site

Contacteur l'auteur (/page-contact)

---

Apprendre programmation python - django web