

## Convert List of Dicts to DOM Tree

**Author :** Frederic De Zorzi  
**Contact :** [fredz@nospam.pimentech.fr](mailto:fredz@nospam.pimentech.fr)  
**Revision :** 1.2  
**Date :** 2009-12-02  
**Copyright :** This document has been placed in the public domain.  
**Tags :** python dom array tree

### Sample

```
In [1]:from libcommon.list2dom import list2dom
In [2]:l = [ {'region':'BRETAGNE', 'departement':'Ille-et-Vilaine', 'uid':'183723', 'nom':'AGENCE INTERNATIONALE MERCURE BRETAGNE - PAYS DE LOIRE - BASSE NORMANDE'},
             {'region':'BRETAGNE', 'departement':'Morbihan', 'uid':'183723', 'nom':'AGENCE INTERNATIONALE MERCURE BRETAGNE - PAYS DE LOIRE - BASSE NORMANDE'},
             {'region':'MONACO', 'departement':'', 'uid':'183723', 'nom':'AGENCE INTERNATIONALE MERCURE BRETAGNE - PAYS DE LOIRE - BASSE NORMANDE'}
in [3]:dom = list2dom(l, ['region'], ['departement'], ['uid', 'nom'])
in [4]:dom.toxml()
out [4]:
<?xml version="1.0" ?>
<root>
  <region region="BRETAGNE">
    <departement departement="Ille-et-Vilaine">
      <uid nom="AGENCE INTERNATIONALE MERCURE BRETAGNE - PAYS DE LOIRE - BASSE NORMANDE">
    </departement>
    <departement departement="Morbihan">
      <uid nom="AGENCE INTERNATIONALE MERCURE BRETAGNE - PAYS DE LOIRE - BASSE NORMANDE">
    </departement>
  </region>
  <region region="MONACO">
    <departement departement="">
      <uid nom="AGENCE INTERNATIONALE MERCURE BRETAGNE - PAYS DE LOIRE - BASSE NORMANDE">
    </departement>
  </region>
</root>
In [5]:dom.childNodes[0].childNodes
Out [5]:[<DOM Element: region at 0xb7890c2c>, <DOM Element: region at 0xb7890dcc>]
In [6]:e=dom.childNodes[0].childNodes[0]
In [7]:e.getAttribute('region')
Out [7]:'BRETAGNE'
In [8]:e['region']
Out [8]:'BRETAGNE'
```

### Source

The source is located in pimentech libcommonPython package : <http://www.pimentech.fr/en/technologies/outils> :you can also download it here : <ftp://ftp.pimentech.net/src/libcommonPython/src/python/libcommon/list2dom.py>

```

from xml.dom import minidom

def __my_getitem(self, key):
    a = self.getAttributeNode(key)
    if not a:
        raise KeyError, key
    return a.value

minidom.Element.__getitem__ = __my_getitem

def list2dom(list_dict, *args, **kwargs):
    """
    list_dict : list of dictionaries
    *args : [ KEY_level_1, supp_key1_level_1, .. supp_keyn_level_1 ], .. [ KEY_level_1, .. ]
    returns :
    (mini)dom tree
    """
    document = minidom.Document()
    tree = document.createElement('root')
    document.appendChild(tree)

    def insert_elements(root_tree, tree, elements):
        if len(elements) == 0:
            return
        element = elements.pop()
        id_key = element['id_key']
        found = False
        for el in tree.childNodes:
            if el.getAttribute(id_key) == str(element[id_key]):
                found = True
                insert_elements(root_tree, el, elements)
                break
        if not found:
            el = root_tree.createElement(id_key)
            for key, value in element.items():
                if key != 'id_key':
                    el.setAttribute(key, str(value))
                    tree.appendChild(el)
            insert_elements(root_tree, el, elements)

    for dict in list_dict:
        elements = []
        for keys in args:
            element = {}
            for i in range(len(keys)):
                key = keys[i]
                if i == 0:
                    element['id_key'] = key
                element[key] = dict[key]
            elements.append(element)
        elements.reverse()
        insert_elements(document, tree, elements)
    return document

```