

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
#
#The MIT License
#
#Copyright (c) 2011
#
#Permission is hereby granted, free of charge, to any person obtaining a copy
#of this software and associated documentation files (the "Software"), to deal
#in the Software without restriction, including without limitation the rights
#to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
#copies of the Software, and to permit persons to whom the Software is
#furnished to do so, subject to the following conditions:
#
#The above copyright notice and this permission notice shall be included in
#all copies or substantial portions of the Software.
#
#THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
#IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
#FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
#AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
#LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
#OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
#THE SOFTWARE.
#
#Authors:
#   Vili Auvinen (vili.k.auvinen@jyu.fi)
#   Olli Kauppinen (olli.kauppinen@jyu.fi)
#   Juho Tammela (juho.i.tammela@jyu.fi)

''' The module is for getting Microsoft Office file formats' docx, pptx, and xlsx meta informations.
The module is not currently used, but it will be in future development.

@author: Juho Tammela
'''

import sys
import zipfile
from xml.dom import minidom

#   def __init__(self, filename):
#       '''
#       filename -- the name or path of the document-file.
#       '''
#       self.filename = filename
#
#       if zipfile.is_zipfile(filename):
#           mydocument = zipfile.ZipFile(filename)
#           coreFile = mydocument.read('docProps/core.xml')
#           self.coreXml = minidom.parseString(coreFile)

def getElementValue(elementTagName, coreXml):
    ''' Returns the text content of the given element.

    elementTagName -- the tag name of the element.

    @return: The text content of the element. If no text content is found, returns the error message.
    '''
    if (coreXml):
        value = ""
        try:
            value = coreXml.getElementsByTagName(elementTagName)[0].firstChild.nodeValue
        except IndexError:
            return elementTagName + " - tag not found"
        return value

def getCreator(coreXml):
    ''' Return the creator of the document.'''
```

```
    return getElementValue('dc:creator', coreXml)

def getLastModifier(coreXml):
    '''Return the last modifier'''
    return getElementValue('dcterms:created', coreXml)

def getCreateDate(coreXml):
    return getElementValue('dcterms:created', coreXml)

def getLastModifiedDate(coreXml):
    return getElementValue('dcterms:modified', coreXml)

def getRevision(coreXml):
    return getElementValue('cp:revision', coreXml)

if __name__ == "__main__":

    filename = 'sampleFiles/docx/Teija_Holtta.docx'

    if zipfile.is_zipfile(filename):
        mydocument = zipfile.ZipFile(filename)
        coreFile = mydocument.read('docProps/core.xml')
        docxCoreXml = minidom.parseString(coreFile)

#    docxMetaReader = MsoMeta("docx_demo.docx")
#    print docxMetaReader.filename
    print 'Creator: ' + getCreator(docxCoreXml)
    print 'Last modified by: ' + getLastModifier(docxCoreXml)
    print 'Created: ' + getCreateDate(docxCoreXml)
    print 'Modified: ' + getLastModifiedDate(docxCoreXml)
    print 'Revision: ' + getRevision(docxCoreXml)
```