```
#!/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 conversions. There is functions for convert the  measure to right dimension.

@author: Vili Auvinen, Olli Kauppinen, Juho Tammela
'''

def convertCmOrInToPt(value):
    '''Converts a value in centimeters or inches to points.

    @param value: the string contains a value centimeter or inches with 'cm' or 'in' after the value

    @note: XML-example:
    <style:paragraph-properties fo:margin-top="0.39cm" </style:paragraph-properties>

    @return: converted value in points (pt).
    @rtype: string

    '''

    try:
        if value.endswith('cm'):
            str(value)
            cmSplit = value.split('cm')
            return str(round(float(cmSplit[0]) * 28.3464567, 0))
        elif value.endswith('in'):
            str(value)
            cmSplit = value.split('in')
            return str(round(float(cmSplit[0]) * 72, 0))
        elif value == "0":
            return '0.0'
    except:
        return

def convertPtToCm(pt):
    return pt * 0.0352777778

def convertPtToTwip(pt):
    return pt * 20.0
```

```
def convertTwipToPt(twips):
    return twips / 20.0

def convertCmToTwip(cm):
    return cm * 566.929133858

def convertTwipToCm(twips):
    return twips * 0.001763889

def convertTwipToInch(twips):
    return twips * 0.000694444

def convertInchToTwip(inches):
    return inches * 1440

def convertPercentToDecimal(percent):
    '''Converts a percent to a decimal format.

    @return: The value in decimal format.
    '''
    str(percent)
    if percent.endswith('%'):
        return str(round(float(percent.split('%')[0]) / 100.0, 1))
    else:
        return percent


def convertCmOrInToString(value, decimals=1):
    '''Converts the value in centimeters or inches to the string format.

    @param value: the string contains centimeter or inches with 'cm' or 'in' after value.
    @param decimals: the number of the decimals (default 1).

    @return: The value in centimeters.
    @rtype: string
    '''
    value = str(value)
    if value.endswith('cm'):
        cmSplit = value.split('cm') #'cm' deleted
        return str(round(float(cmSplit[0]), decimals))
    elif value.endswith ('in'):
        inSplit = value.split('in') #'in' deleted
        return str(round(float(inSplit[0]) * 2.54, decimals))
    else:
        return str(float(value))



def convertCmOrInDictToString(convertedDict, decimals=1):
    '''Converts dictionary's value in centimeters or inches to the string format.

    @param convertedDict: The dictionary contains centimeter or inches with 'cm' or 'in' as values.
    @param decimals: the number of the decimals (default 1).

    @return: Converted dictionary in centimeters.
    '''
    for key in convertedDict.keys():
        value = convertedDict[key]

        if str(value).endswith('cm'):
            value = value.split('cm')[0] #'cm' deleted
        elif str(value).endswith('in'):
            value = value.split('in')[0] #'in' deleted
            value = float(value) * 2.54
        value = round(float(value), decimals)
        convertedDict[key] = str(value)
    return convertedDict
```