

Conversion de températures

```
<sxh python; title : convertisseur_temperature.py> #!/usr/bin/env python # -*- coding: utf-8 -*- #
Conversion de températures # programme réalisé par AC&JD, ba2 chimie 2010-2011
```

```
from Tkinter import *
```

```
def delwidgets():
```

```
    for widget in F.winfo_children():
        if isinstance(widget, Toplevel):
            widget.destroy()
```

```
def affiche(titre, reponse): # Permet de n'afficher qu'un seul Toplevel (Fenêtre affichant les réponses)
avec un destroy des précédents #
```

```
    for widget in F.winfo_children():
        if isinstance(widget, Toplevel):
            widget.destroy()
    #####
    conteneur = Toplevel(F) # Création de la fenêtre de réponse #
    conteneur.title('Résultat') # Création du titre de la fenêtre #
    Label(conteneur, text=titre).pack() # Label défini un fragment de texte
qui peut être utilisé pour afficher des informations dans la fenêtre
"conteneur" #
    Label(conteneur, text='Résultat: ' + str(reponse)).pack() # Pack permet
d'adapter la géométrie de la fenêtre au texte. "str" défini une chaîne de
caractères #
    Button(conteneur, text='Ok', command=conteneur.destroy).pack(padx=5,
pady=5) # Création du bouton qui détruit la fenêtre et définition de la
grandeur du bouton #
```

```
def erreur(texte):
```

```
    delwidgets()
    erreurwidget = Toplevel(F)
    erreurwidget.title('Erreur')
    Label(erreurwidget, text=texte).pack(padx=5, pady=5)
    Button(erreurwidget, text='Ok',
command=erreurwidget.destroy).pack(padx=5, pady=5)
```

```
# Création des fonctions permettant de convertir les températures, les formules ont été prises sur
wikipedia # def CelToFah(celsius) :
```

```
    fahrenheit = 9./5.*int(celsius)+32
    affiche('Conversion de Celsius en Fahrenheit', fahrenheit)
```

```
def CelToKel(celsius) :
```

```
kelvin = int(celsius)+273.15
affiche('Conversion de Celsius en Kelvin', kelvin)
```

def FahToCel(fahrenheit) :

```
celsius = 5./9.*(int(fahrenheit)-32)
affiche('Conversion de Fahrenheit en Celsius', celsius)
```

def KelToCel(kelvin) :

```
if int(kelvin)<0 :
    erreur('ne peut être inférieur à 0')
else:
    celsius = int(kelvin)-273.15
    affiche('Conversion de Kelvin en Celsius', celsius)
```

def FahToKel(fahrenheit) :

```
kelvin = 5./9.*(int(fahrenheit)+459.67)
affiche('Conversion de Fahrenheit en Kelvin', kelvin)
```

def KelToFah(kelvin) :

```
fahrenheit = 9./5.*(int(kelvin)-459.67)
affiche('Conversion de Kelvin en Fahrenheit', fahrenheit)
```

F = Tk() # Création de la classe Tk() qui permet d'engendrer différents types de fenêtres, de modifier les tailles,... # w = Frame(F, height=500, width=500) # Création de la fenêtre de travail et définition des dimensions # w.pack()

```
# Bouton permettant la destruction de la fenêtre et définition de sa position dans la fenêtre # ##
Création de boutons de conversion. Ceux-ci vont exploiter la valeur qui sera entrée par l'utilisateur
(entree.get) et la fonction correspondant au bouton ## ### On utilise la fonction lambda pour
pouvoir utiliser un argument dans la fonction command ### Button(F, text='Quitter', anchor=SE,
command=F.quit).pack(side=BOTTOM, anchor=SE) Button(F, text='CelToFah', anchor=SE,
command=lambda: CelToFah(entree.get())).pack(side=BOTTOM, anchor=SW) Button(F,
text='CelToKel', anchor=SW, command=lambda: CelToKel(entree.get())).pack(side=BOTTOM,
anchor=SW) Button(F, text='KelToFah', anchor=SW, command=lambda:
KelToFah(entree.get())).pack(side=BOTTOM, anchor=SW) Button(F, text='KelToCel', anchor=SW,
command=lambda: KelToCel(entree.get())).pack(side=BOTTOM, anchor=SW) Button(F,
text='FahToCel', anchor=SW, command=lambda: FahToCel(entree.get())).pack(side=BOTTOM,
anchor=SW) Button(F, text='FahToKel', anchor=SW, command=lambda:
FahToKel(entree.get())).pack(side=BOTTOM, anchor=SW)
```

Quelques textes, affichés dans la fenêtre et entrés de la même façon que précédemment

```
Label(w,text="Convertisseur de température").pack()
Label(w,text="=====").pack() Label(w, text = 'Entrez la
Température :', anchor=SW).pack(anchor=SW) # Anchor permet de positionner le texte où l'on veut
```

dans la fenêtre # entree = Entry(w) # Permet à l'utilisateur d'entrer une valeur qui sera exploitée grâce aux boutons de conversion # entree.pack(anchor=SW)

F.mainloop() # Permet de faire tourner le programme en boucle en attendant qu'il soit utilisé # </sxh>

From:

<https://dvillers.umons.ac.be/wiki/> - **Didier Villers, UMONS - wiki**

Permanent link:

https://dvillers.umons.ac.be/wiki/teaching:progappchim:conversion_temperature_2011

Last update: **2014/02/11 08:16**

