

```
# Programm:      Temperaturen
#
# Datum:         11.06.2019
#
# Autor:         Stefan Duscher
#
# Bemerkungen:   Beispiel VHS-Kurs zum Einsatz eines GUI (graphic user interface)
#
# -----

# Importieren der Bibliothek tkinter
# Diese Bibliothek stellt alle Funktionen zur Verfügung, um mittels
# Fenstern mit dem Bediener zu kommunizieren (Ein- und Ausgabe)
#
from tkinter import * # Komplette Importierung der Bibliothek tkinter

# Festlegen welchen Wert der absolute Nullpunkt in den einzelnen Einheiten hat
# Dies sind globale Variablen; da sie nicht geändert werden sind sie Konstanten
#
ABSOLUTER_NP_K = 0.0      # absoluter Nullpunkt in Kelvin
ABSOLUTER_NP_C = -273.15  # absoluter Nullpunkt in Celsius
ABSOLUTER_NP_F = -459.67  # absoluter Nullpunkt in Fahrenheit

# Weitere notwendige Konstanten für die Umrechnung
#
NULL_F = 32.0            # 0° C in Fahrenheit
FAKTOR_F_C = 9/5         # Umrechnungsfaktor zwischen Fahrenheit und Celsius

# Definition eines Fehlertextes, wenn eine unmögliche Temperatur eingegeben wird
# Unmöglich ist Temperatur unterhalb des absoluten Nullpunktes
#
FEHLERMELDUNG_TEMP = "**** Fehler: Unmögliche Temperatur! ****"

# Funktion, die in Abhängigkeit von der Auswahl der Berechnung den Antwortstring
# erzeugt und zurückgibt
#
def auswahl_knopfdruck():
    temperatur_str = eingabefeld.get() # Auslesen der Eingabe als String
    temperatur = float(temperatur_str) # Umwandeln der Eingabe in eine
    Fließkommazahl (float)
    wahl = variable.get() # Auslesen, welche Umrechnung gewählt ist

    # Erzeugen des Ausgabestrings
    if wahl == u1:
        message = temperatur_str + "°C = " + str(Celsius_Kelvin(temperatur)) + "K"
    elif wahl == u2:
        message = temperatur_str + "°C = " + str(Celsius_Fahrenheit(temperatur)) + "F"
    elif wahl == u3:
        message = temperatur_str + "K = " + str(Kelvin_Celsius(temperatur)) + "°C"
    elif wahl == u4:
        message = temperatur_str + "K = " + str(Kelvin_Fahrenheit(temperatur)) + "F"
```

```
elif wahl == u5:
    message = temperatur_str + "F = " + str(Fahrenheit_Celsius(temperatur)) + "°C"

elif wahl == u6:
    message = temperatur_str + "F = " + str(Fahrenheit_Kelvin(temperatur)) + "K"

ausgabe.configure(text=message) # Verändern des Labels "ausgabe", indem ihm der Text "messsage" zum Anzeigen gegeben wird

# Funktion zur Umrechnung von Celsius nach Kelvin
def Celsius_Kelvin(t):
    if t >= ABSOLUTER_NP_C: # Berechnen, wenn Temperatur größer oder gleich dem absoluten Nullpunkt
        return t - ABSOLUTER_NP_C
    else: # andernfalls Fehlermeldung ausgeben
        raise TypeError(FEHLERMELDUNG_TEMP)

# Funktion zur Umrechnung Celsius nach Fahrenheit
def Celsius_Fahrenheit(t):
    if t >= ABSOLUTER_NP_C: # Berechnen, wenn Temperatur größer oder gleich dem absoluten Nullpunkt
        return NULL_F + FAKTOR_F_C*t
    else: # andernfalls Fehlermeldung ausgeben
        raise TypeError(FEHLERMELDUNG_TEMP)

# Funktion zur Umrechnung von Kelvin nach Celsius
def Kelvin_Celsius(t):
    if t >= ABSOLUTER_NP_K: # Berechnen, wenn Temperatur größer oder gleich dem absoluten Nullpunkt
        return t + ABSOLUTER_NP_C
    else: # andernfalls Fehlermeldung ausgeben
        raise TypeError(FEHLERMELDUNG_TEMP)

# Funktion zur Umrechnung von Kelvin nach Fahrenheit
def Kelvin_Fahrenheit(t):
    if t >= ABSOLUTER_NP_K: # Berechnen, wenn Temperatur größer oder gleich dem absoluten Nullpunkt
        return t*FAKTOR_F_C + ABSOLUTER_NP_F
    else: # andernfalls Fehlermeldung ausgeben
        raise TypeError(FEHLERMELDUNG_TEMP)

# Funktion zur Umrechnung von Fahrenheit nach Celsius
def Fahrenheit_Celsius(t):
    if t >= ABSOLUTER_NP_F: # Berechnen, wenn Temperatur größer oder gleich dem absoluten Nullpunkt
        return (t - NULL_F)/FAKTOR_F_C
    else: # andernfalls Fehlermeldung ausgeben
        raise TypeError(FEHLERMELDUNG_TEMP)
```

# Funktion zur Umrechnung von Fahrenheit nach Kelvin

```
def Fahrenheit_Kelvin(t):
    if t >= ABSOLUTER_NP_F:      # Berechnen, wenn Temperatur größer oder gleich dem
        absoluten Nullpunkt
        return (t - ABSOLUTER_NP_F)/FAKTOR_F_C
    else:                        # andernfalls Fehlermeldung ausgeben
        raise TypeError(FEHLERMELDUNG_TEMP)
```

```
#
#
# Hier beginnt das Hauptprogramm
# -----
#
```

```
fenster = Tk() # fenster die Eigenschaft eines Objektes der Bibliothek Tk zuweisen
               # Tk() ist Bestandteil der Bibliothek tkinter
```

```
fenster.title("Temperatur Umwandler") # Dem Fenster eine Überschrift / einen
    Titel zuweisen
                                     # Oben links in Kopfzeile des Fensters
```

```
# Definition der Beschreibung, die an oberster Stelle im Hauptfenster stehen soll
info = Label(fenster, justify = LEFT, font=("Helvetica", 16),
             text="""\
```

```
***** TEMPERATUR UMWANDLER *****
***** VHS-Kurs Python für Anfänger *****
```

- 1) Gewünschte Umrechnung wählen.
- 2) Temperatur eingeben.
- 3) Taste "Umrechnen" drücken."""

```
# Definition, welche Umrechnungen zur Auswahl stehen sollen und welche Texte
# hierbei bei der Auswahl angezeigt werden sollen
#
```

```
u1 = "von Celsius nach Kelvin"
u2 = "von Celsius nach Fahrenheit"
u3 = "von Kelvin nach Celsius"
u4 = "von Kelvin nach Fahrenheit"
u5 = "von Fahrenheit nach Celsius"
u6 = "von Fahrenheit nach Kelvin"
```

```
# Definition eines Strings, der im Fenster angezeigt werden soll
# Hier kann nicht einfach ein beliebiger String gewählt werden, sondern er muss
# vom Typ her zur Klasse der tk.StringVar gehören
#
```

```
variable = StringVar(fenster)
variable.set(u1) # Definition des Wertes von variable, wenn man Programm
                startet; der Default-Wert
```

```
# Definition der Optionen, die angezeigt werden, wenn man auf das Feld mit der
    Umrechnung klickt
```

```
optionen = OptionMenu(fenster, variable, u1, u2, u3, u4, u5, u6) # Es gehört zu
    Objekt "fenster", angezeigt werden soll
```

```
# etwas, das in
    "variable" liegt und u1 bis u6 können
```

```
liegen # in "variable"

optionen.configure(width = 40, font=("Helvetica", 16)) # Definition der Breite
und Schriftart des Feldes # wo die Art der

Umrechnung gewählt wird

eingabefeld = Entry(fenster, bd=5, width=20) # Definition des Eingabefeldes: Es
gehört zu Objekt "fenster", hat einen Rand # und die Breite des Feldes wird
definiert

label = Label(fenster, justify = LEFT, font=("Helvetica", 16), text="Ausgabe: ") # Definition des Labels, wo das Ergebnis
# ausgegeben wird; ein Label kann nur ausgeben

ausgabe = Label(fenster, justify = CENTER, font=("Helvetica", 16), text="Noch
nichts berechnet") # Definition, was als Ergebnis angezeigt werden soll,
# wenn Programm gestartet wird

run_button = Button(fenster, text="Umrechnen", font=("Helvetica", 16), command =
auswahl_knopfdruck) # Definition des Buttons, mit dem man das Umrechnen auslöst
# "Button()" hat als Rückgabe, ob es gedrückt wurde oder nicht

info.grid(row = 0, column = 0, columnspan = 3, padx = 50, pady = 20) #
Definition, wo die Beschreibung im Fenster angezeigt werden soll #

optionen.grid(row = 1, column = 0, columnspan = 3, pady = 20) # Definition, wo
der Auswahlknopf der Umrechnungsart angeordnet werden soll #

eingabefeld.grid(row = 2, column = 0, pady = 20) # Definition, wo das
Eingabefeld angeordnet werden soll #

run_button.grid(row = 2, column = 1, pady = 20) # Definition, wo der Knopf zum
Auslösen angeordnet werden soll #

label.grid(row = 3, column = 0, pady = 20) # Definition, wo das Label, das
"Ausgabe: " anzeigt, angeordnet werden soll #

ausgabe.grid(row = 3, column = 1, columnspan = 2, pady = 20) # Definition, wo
das Rechenergebnis angezeigt werden soll #

fenster.mainloop() # Definition des Programmes als Schleife, d.h. das
Hauptprogramm wird solange wiederholt,
# bis es jemand beendet, z.B. durch Schließen des Fensters
```