

Echelles de température

Historiquement, la mesure de la température s'est faite grâce à la dilatation des corps liquides. D'abord qualitative, cette mesure s'est progressivement affinée et est devenue quantitative par la mise en place d'échelles de mesure basées sur des points de référence fixes, tels que la fusion de la glace.

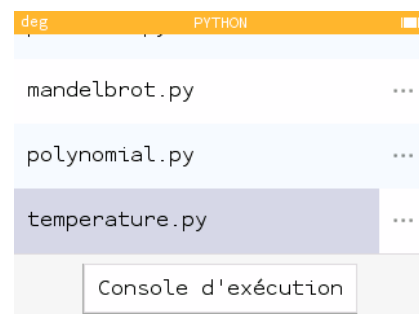
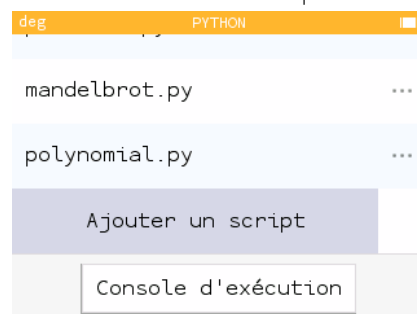
Les échelles les plus connues sont le degré Celsius, le degré Fahrenheit et le kelvin.

Vers le kelvin

Notre échelle de référence, le degré Celsius, admet deux points de référence fixe : le point de fusion de l'eau et son point d'ébullition, séparés d'exactement 100 parts égales, correspondant donc respectivement à 0°C et 100°C. C'est l'échelle de mesure que nous utilisons dans notre quotidien.

En sciences cependant, l'unité de température du système international est le kelvin, une unité dont les intervalles sont semblables à ceux du degré. En revanche, ces deux échelles sont décalées, de façon à ce que 0K est équivalent à -273,15°C.

1. A combien de kelvin équivaut une température de 0°C ? Et une température de 100°C ?
2. Soit x la température d'un corps exprimée en degrés Celsius, et y la température de ce même corps exprimée en kelvin. Exprimer y en fonction de x . De quel type de fonction s'agit-il ?
3. Créons dans l'application Python de notre calculatrice un nouveau script intitulé **temperature.py**. Nous y écrirons les différentes fonctions que nous allons réaliser dans cet exercice.



On souhaite écrire dans ce script une fonction **CenK(x)** permettant de retourner, lorsqu'on entre dans le programme une température en degrés Celsius x , la température en kelvin correspondante.

Une fonction est définie en Python à l'aide de l'instruction : **def nom_de_la_fonction(variable1, variable2, ...)** : Les instructions qui sont à exécuter par la fonction doivent être indentées, c'est-à-dire décalées par rapport au début de la ligne, ce que la calculatrice réalise spontanément lorsqu'on revient à la ligne avec la touche EXE.

Ici, nous allons demander dans un premier temps à notre programme de calculer y , la température en kelvin, puis ensuite d'en faire le résultat de notre fonction à l'aide de l'instruction **return**.

La commande **return** permet d'associer une variable de sortie à l'exécution de la fonction. Attention,

cette commande arrête l'exécution de celle-ci, il faut donc que ce soit la toute dernière ligne de notre programme.

Compléter le programme afin qu'il réalise l'objectif mentionné :

```
1 def CenK(...):  
2     y = x + ...  
3     return ...
```

4. On cherche maintenant à faire l'inverse.

- (a) Ecrire la fonction **KenC(y)** permettant de retourner, pour une température en kelvin y donnée, la température en degrés Celsius.
- (b) Tester le programme pour $T = 32^\circ\text{C}$.

Vers le degré Fahrenheit

Si historiquement le degré Fahrenheit est antérieur aux deux autres échelles dont nous avons déjà parlé, son utilisation est plutôt restreinte aux pays d'Amérique du nord.

Les températures de référence utilisées sont différentes, de même que l'intervalle utilisé. Ainsi c'est la température du corps humain qui sert à définir les 100°F . Dans cette échelle, l'eau gèle à 32°F et bout à 212°F .

- 1. On appelle g la fonction qui, à x température en degrés Celsius, associe une température y en degrés Fahrenheit. Exprimer les données de l'énoncé relatives aux changements d'état de l'eau à l'aide de cette fonction.
- 2. On admet que la fonction g est une fonction affine. A l'aide de la question précédente, déterminer son coefficient directeur et son ordonnée à l'origine, et en déduire une expression de la fonction $g(x)$.
- 3. Ecrire la fonction **CenF(x)** qui, appliquée à une température donnée x en degrés Celsius, retourne la température en degrés Fahrenheit.
- 4. On souhaite utiliser les fonctions précédentes afin d'écrire une fonction **KenF(t)** (qui permet de transformer les degrés Fahrenheit en kelvin) sans réaliser aucun calcul.
 - (a) Il est possible, à l'intérieur d'une fonction, d'appeler une autre fonction déjà définie. Compléter l'algorithme suivant :

```
1 def KenF(t):  
2     degrees = KenC(...)  
3     fahr = CenF(...)  
4     return ...
```

- (b) Tester le programme pour $T = 310^\circ\text{K}$.

On pourra modifier le programme afin que le résultat soit arrondi à l'entier grâce à la commande **round(valeur)**.