

Les boucles

1) La boucle while (tant que)

Le programme que nous avons écrit précédemment comporte un défaut. Si un utilisateur entre une distance ou une durée négative, le programme affichera quand même une valeur de vitesse, mais qui n'a pas de sens pour son utilisateur. L'instruction **while** va permettre de vérifier la bonne saisie de l'utilisateur. Voici ce que cela donne pour la saisie de la distance :

```
distance=-0.1
```

```
while distance<0:
```

```
    distance=float(input("d en km "))
```

La structure est la suivante :

Instruction d'initialisation de la variable testée pour rendre le test vrai

```
while test:
```

```
    instruction
```

La variable test est appelée variable logique ou Booléenne. Elle ne peut prendre que deux valeurs 1 (true) ou 0 (false). Ainsi, en écrivant **distance<0**, une variable logique est créée en mémoire RAM à l'exécution. Tant que son contenu est 1, c'est-à-dire qu'elle est vraie, l'instruction de la boucle est exécutée.

Si l'utilisateur entre -5 pour la distance, cette dernière lui est demandée à nouveau jusqu'à ce qu'il entre une valeur adéquate.

En revanche si l'utilisateur se trompe de format et entre 5,2 au lieu de 5.2, c'est un message d'erreur qui apparaît en rouge à l'exécution et qui indique le numéro de l'instruction qui pose problème ainsi que le type d'erreur commise, ce qui est une aide précieuse pour corriger cette erreur de saisie pour un utilisateur ou de codage pour un programmeur.

Le programme complet qui demande une saisie de distance et de durée puis affiche un message sur la façon dont vous roulez est ainsi :

```
distance=-0.1
```

```
while distance<0:
```

```
    distance=float(input("d en km "))
```

```
duree=-0.1
```

```
while duree<0:
```

```
    duree=float(input("t en h "))
```

```
vitesse=distance/duree
```

```
if vitesse<50:
```

```
    print("vous roulez trop lentement")
```

elif vitesse>120:

print("vous roulez très dangereusement")

elif vitesse>90:

print("vous roulez trop vite")

Il est conseillé de sauvegarder les fichiers source de Python en faisant un copié collé de ces dernières dans un fichier bloc note par exemple. Le même copié collé en sens inverse permettra de réutiliser le programme dans une nouvelle session de Python en ligne par exemple.

Remarque :

Le programme précédent présente un inconvénient. Si l'utilisateur entre pour la distance ou la durée une chaîne de caractère comme A, alors un message d'erreur apparaît. Il convient donc d'ajouter quelques lignes de programme pour tester si la saisie de l'utilisateur est une chaîne de caractère convertible en un nombre décimal, grâce à la fonction **str.isdigit(chaine)** qui renvoie la valeur logique

True si la variable chaîne est convertible en nombre à virgule et False dans le cas contraire. Voilà le programme modifié

distance=-0.1

d="A"

while distance<0 and not str.isdigit(d):

d= input("d en km ")

if str.isdigit(d):

distance=float(d)

else:

print("entrez un nombre positif pour la distance")

duree=-0.1

t="A"

while duree<0 and not str.isdigit(t):

t=input("t en h ")

if str.isdigit(t):

duree=float(t)

else:

print("entrez un nombre positif pour la durée")

vitesse=distance/duree

if vitesse<50:

```
print("vous roulez trop lentement")
```

```
elif vitesse>120:
```

```
    print("vous roulez très dangeusement")
```

```
elif vitesse>90:
```

```
    print("vous roulez trop vite")
```

```
elif 1==1 :
```

```
    print("vous roulez à la bonne vitesse")
```