



Activité 1 : Découverte de la boucle `while` | 

1. Sur votre ordinateur, lancer le logiciel *Thonny*.
2. Dans la console entrer et exécuter la commande suivante qui permet de charger en mémoire la fonction `randint`.

```
>>> from random import randint
```

3. La fonction `randint(a, b)` renvoie un nombre entier choisi au hasard entre deux entiers `a` et `b`. Par exemple la commande suivante permet de simuler le lancer d'un dé à six faces.

```
>>> randint(1, 6)
```

Exécuter la commande précédente plusieurs fois pour visualiser la variation aléatoire des nombres obtenus. *Rappel.* Pour naviguer dans l'historique des commandes dans la console, utiliser les touches  et .

4. Dans *Thonny*, créer un nouveau fichier de programme appelé `lancers_de.py` et enregistrez-le dans votre dossier `SNT`.
  - (a) Recopier dans *Thonny* et exécuter le programme ci-dessous plusieurs fois.

```
1 from random import randint
2
3 face = randint(1, 6)
4 print(face)
5 essais = 1
6 while face < 6:
7     print("Raté !")
8     face = randint(1, 6)
9     print(face)
10    essais = essais + 1
11 print("Gagné en", essais, "lancers !")
```

- (b) Ce programme simule un jeu de dé. Expliquer les règles de ce jeu. *Aide.* Le mot clé python `while` se traduit en français par « tant que ».
- (c) Dans le programme, à la ligne 3, remplacer `randint(1, 6)` par `6`. Exécuter plusieurs fois le programme après ce changement. Que constatez-vous ? Utiliser le mode débogueur de *Thonny* pour l'expliquer.
- (d) Dans le programme, réécrire la ligne 3 dans sa version initiale. Ensuite, à la ligne 8, remplacer `randint(1, 6)` par `2`. Que constatez-vous ? Utiliser le mode débogueur de *Thonny* pour l'expliquer.

Définition 1 : Boucle `while`

Une boucle `while` permet de répéter une (ou plusieurs) instruction(s) tant qu'une condition est vraie. La boucle s'arrête dès que la \_\_\_\_\_ devient \_\_\_\_\_.

### Propriété 1 : Boucle `while`, les trois cas possibles

Lorsqu'une boucle `while` est rencontrée dans un programme, trois cas différents peuvent se présenter :

1. La \_\_\_\_\_ est vraie avant la boucle donc on entre dans la boucle.
  - (a) *Cas 1.* Au bout d'un certain nombre de répétitions de la boucle, la condition devient \_\_\_\_\_ donc on sort de la boucle et le programme continue. On dit que la **la boucle se termine**.
  - (b) *Cas 2.* Au cours de l'exécution de la boucle, la condition reste toujours \_\_\_\_\_. On reste bloqué dans la boucle. On dit que la boucle \_\_\_\_\_.
2. *Cas 3.* La \_\_\_\_\_ est \_\_\_\_\_ avant la boucle donc on n'entre pas dans la boucle. La boucle n'est pas exécutée et le programme continue.

### Activité 2 : Boucle `while` avec *Py-Rates* |

Retourner dans *Py-Rates* (<https://py-rates.fr>). Réaliser les étapes suivantes. En cas de blocage, solliciter le professeur.

1. Reprenez votre partie à l'aide de votre identifiant.
2. Déplacer le curseur de vitesse tout à droite côté lièvre.
3. Réaliser le niveau 8. Pour cela :
  - (a) Lire les consignes dans le haut du panneau de gauche.
  - (b) Consulter la rubrique *Boucle while* dans le menu d'aide à gauche.
  - (c) En vous inspirant de l'aide, compléter le programme afin de prendre la clé puis atteindre ensuite et ouvrir le coffre. Si nécessaire, régler la vitesse.

### Activité 3 : Jeu du nombre mystère |

1. Dans *Thonny*, créer un nouveau fichier de programme appelé `jeu_mystere.py` et enregistrez-le dans votre dossier `SNT`. Recopier ensuite et exécuter le programme ci-dessous. *Remarque.* L'opérateur `!=` signifie « différent de ».

```

1  from random import randint
2
3  nbre_ordi = randint(1, 10)
4  nbre_joueur = 0
5  print("J'ai choisi un nombre entier entre 1 et 10. Devinez-le !")
6  while nbre_joueur != nbre_ordi:
7      nbre_joueur = int(input("Faites un essai : "))
8      if nbre_joueur != nbre_ordi:
9          print("Raté !")
10 print("Gagné !")

```

2. Dans *Thonny* est-il possible de tricher à ce jeu ? Expliquer.
3. Modifier le programme pour que le nombre choisi par l'ordinateur soit entre 1 et 100. Comment modifier le programme pour qu'il affiche `"Vous chauffez !"` si le joueur s'approche du nombre à deviner ou `"Vous refroidissez !"` sinon.