

BOUCLES ET ITÉRATEURS EN PYTHON

Pour lancer un dé en Python, on peut faire

```
from random import *
print(randint(1,6))
```

mais avec cette méthode, même s'il n'est pas nécessaire de réimporter les méthodes de l'objet *random*, lancer le dé 20 fois est pénible.

1. Boucler 20 fois

Essayer le script suivant :

```
from random import *
for compteur in range(20):
    print(randint(1,6))
```

Combien de fois le dé a-t-il été lancé en tout ?.....

Essayer la variante suivante :

```
from random import *
for compteur in range(1,20):
    print(randint(1,6))
```

Cette fois-ci, combien de fois le dé a-t-il été lancé ?.....

2. Tests sur le compteur

En fait, *compteur* est une variable qui change de valeur à chaque passage dans la boucle. On peut s'en convaincre en l'affichant :

```
for compteur in range(5):
    print('Maintenant le compteur vaut', compteur)
```

Quelles sont les valeurs successives du compteur ?.....

Compléter le script suivant pour que le compteur prenne successivement les valeurs 1, 2, 3, 4 et 5 :

```
for compteur in range( , ):
    print('Maintenant le compteur vaut', compteur)
```

3. Itérateurs et listes

En Python, on construit une liste en mettant ses éléments entre crochets ; on peut mettre un itérateur dans une liste :

```
print([n for n in range(10)])
```

Compléter le script suivant pour qu'il affiche les carrés des nombres allant de 1 à 10 :

```
print([n**2 for n in range( , )])
```

Dans la suite du TP, on lancera le dé 20 fois en faisant

```
print([randint(1,6) for n in range(20)])
```

Combien de fois le 6 est-il sorti?.....

4. Diagramme en bâtons

Commencer par lancer le dé 100 fois :

```
liste=[randint(1,6) for n in range(100)]
```

Pour obtenir le nombre de fois que le dé est tombé sur 6, on peut faire

```
print(liste.count(6))
```

(ce qui revient à compter le nombre de 6 dans la liste)

Compléter le tableau des effectifs suivant :

Résultats	1	2	3	4	5	6
Effectifs						

En déduire le diagramme des effectifs sur les 100 lancers de dés :

