

Quelles constructions élémentaires sont suffisantes pour écrire n'importe quel programme ?

1 Python : un langage de haut-niveau

- développé par *Guido van Rossum* fin 1989,
- interprété, c'est à dire que le programme est lu ligne après ligne par un *interpréteur*.

2 Constructions élémentaires

2.1 Jeu de dés

Pour construire un tel programme il faut pouvoir :

- Stocker la valeur du dé dans une **variable**.
- Demander à l'utilisateur d'**entrer** une valeur.
- **Répéter** :
 - **Comparer** la valeur de l'utilisateur à celle du dé.
 - Incrémenter le nombre d'essais.
- **Afficher (sortir)** le nombre d'essais dans la console.

2.2 Interpréteur

Il lit le code source ligne par ligne et l'exécute. Il est possible d'écrire du code directement dans la console de l'interpréteur, mais on préférera souvent utiliser un **Environnement de Développement Intégré**.

2.3 Variables

```
1 de1 = 3
```

Code 1 – Une variable est une boîte étiquetée

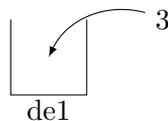


FIGURE 1 – Affectation

2.4 Entrée/Sortie

```
1 proposition =input("Choisir une valeur du dé: ")
```

Code 2 – Entrée : demander une valeur à l'utilisateur

```
1 print(proposition)
```

Code 3 – Sortie : afficher une valeur dans la console

2.5 Comparaison

```
1 if de1 == proposition:
2     print("Gagné")
3 else:
4     print("Perdu")
```

Code 4 – Le mot-clé `if` évalue la comparaison qui le suit.

2.6 Répétition

2.6.1 Boucle non bornée

```
1 de1 = 3
2 proposition = 0
3 while de1 != proposition:
4     proposition = int(input("Choisir: "))
```

Code 5 – Une boucle non bornée (`while`) répète une instruction tant que la condition est vérifiée.

2.6.2 Boucle bornée

```
1 for compteur in range(10):
2     print(compteur)
```

Code 6 – Une boucle bornée (`for`) effectue un nombre d'*itérations* déterminé.

2.7 Bibliothèque

```
1 from random import randint
2 de1 = randint(1,6)
```

Code 7 – La bibliothèque `random` permet de générer un nombre aléatoire.