

La **programmation graphique par blocs** est un excellent point de départ pour apprendre à coder. Elle permet de visualiser la **logique** des programmes et de se familiariser avec les **concepts de base** de la programmation. En comprenant ce **lien** entre les blocs et le texte, on est bien préparé pour aborder les **langages de programmation textuels**.

## Programmation graphique par blocs

- Utilise des blocs visuels pour représenter des instructions.
- Chaque bloc représente une action ou une instruction que l'ordinateur doit exécuter.
- Les blocs sont organisés de manière logique pour former un programme.

### Avantages

Apprentissage relativement facile.  
Intuitif, la manipulation des blocs limite les erreurs de syntaxe.

### Inconvénient

Moins adapté pour des programmes très complexes, mais idéale pour des tâches éducatives de base.

## Programmation textuelle

- Utilise du texte pour écrire des instructions.
- Chaque ligne de code correspond à une action à exécuter.
- Le code est organisé en sections distinctes.

### Avantage

Elle permet plus de complexité, de flexibilité, d'efficacité. Elle est plus adaptée à des projets avancés.

### Inconvénients

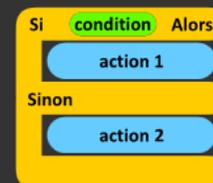
La moindre erreur peut bloquer l'exécution du programme.  
Apprentissage relativement difficile.

### Blocs



### Langage Python

```
for i in range(X):  
    # Action à répéter
```



```
if condition:  
    # Action si la condition est vraie  
else:  
    # Action si la condition est fausse
```

### Exemple : Afficher le score d'un jeu

### Blocs



### Langage Python

```
score = 0  
if score > 10:  
    print("gagné !")  
elif score < 0:  
    print("perdu !")  
else: print("Continue à jouer")
```



image générée par IA



image générée par IA