

Condition et boucle

① ② ③

Écrire, mettre au point et exécuter un programme simple

- Savoir interpréter une condition
- Mettre au point un programme en réponse à un problème donné

1 Situation de recherche

On donne l'algorithme suivant.

```

1:  $k \leftarrow$  nombre aléatoire compris entre 1 et 10
2: SI  $k < 5$  ALORS
3:   DEBUT_SI
4:   Affiche "Le nombre est strictement inférieur à 5"
5:   FIN_SI
6: SINON
7:   DEBUT_SINON
8:   Affiche "Le nombre est supérieur ou égal à 5"
9:   FIN_SINON

```

1. Quelles sont les variables?
2. Explique le rôle de cet algorithme.

Tester le code



Définition 1. Condition

Une **condition** est une instruction qui ouvre le choix parmi deux actions suivant le résultat d'un test. Sa structure est : **Si** test vérifié **alors** Action 1 **sinon** Action 2.



Exemple

Test : il fait beau

Action 1 : je vais à la plage

Action 2 : je vais au cinéma



Syntaxe

```

Si il fait beau Alors
    je vais à la plage
Sinon
    je vais au cinéma
FinSi

```

2 Activité Scratch

On donne l'algorithme suivant.

```

1: VARIABLES
2: y EST_DU_TYPE NOMBRE
3: x EST_DU_TYPE NOMBRE
4: DEBUT_ALGORITHME
5:   LIRE x
6:   SI (x<3) ALORS
7:     DEBUT_SI
8:     y PREND_LA_VALEUR x+3
9:     FIN_SI
10:  SINON
11:    DEBUT_SINON
12:    y PREND_LA_VALEUR x*x
13:    FIN_SINON
14:  AFFICHER y
15: FIN_ALGORITHME

```

Coder cet algorithme.

3 Activité Scratch

Écris un programme qui demande un nombre entier compris entre 0 et 24 et affiche "Buenos dias" si l'utilisateur renseigne un nombre compris entre 0 et 12 ou "Buenas tardes" sinon.

4 Exercice d'application

Recopie et modifier le programme écrit en Python ci-dessous pour qu'il affiche Vert si la lettre V est renseigné.

```

n = input("Taper B ou R")
if n=="B" :
    print("Bleu")
else :
    print("Rouge")

```

5 Activité Scratch

Écrire un programme qui

1. affiche deux nombres entiers aléatoires compris entre -10 et 10
2. demande à l'utilisateur le type s'il souhaite les additionner ou les multiplier
3. affiche le résultat.

7 Activité Scratch

On donne l'algorithme suivant.

- 1: L'ordinateur choisit un nombre entier a aléatoire entre 1 et 1000
- 2: L'ordinateur demande au joueur de donner un nombre entier n entre 1 et 1000
- 3: **TANT QUE** n est différent de a
- 4: **SI** $n < a$ **ALORS**
- 4: Scratch dit "le nombre choisi est trop petit"
- 5: **SINON**
- 4: Scratch dit "le nombre choisi est trop grand"
- 5: **FIN TANT QUE**
- 6: Scratch dit "Bravo, vous avez gagné !"

Coder les deux améliorations du jeu.

1. On souhaite compter le nombre de coups nécessaires par le joueur avant de gagner. Utiliser la variables Score et coder cette nouvelle fonctionnalité.
2. On souhaite conserver les nombres choisis par le joueur.

6 Activité Scratch

1. Dessine un segment dans la section Costume. Utilise le pinceau. Utilise le segment comme lutin.
 2. Anime le segment pour qu'il monte quand la touche **(P)** du clavier est cliquée et qui descend lorsque la touche **(M)** du clavier est cliquée.
- Pour poser des questions à l'utilisateur, il faut utiliser la section **Capteurs**.
 - Pour les directions, on utilise les flèches directionnelles. L'instruction **touche ... pressée?** permet de sélectionner une action liée à une flèche directionnelle.
 - Pour le test, on utilise la section **Contrôle**

Tester le code

