

## Transition Scratch Python en seconde Exemple de progression et de situations

### Situation 1 : un programme de calcul

objectifs relatifs au thème 4 Algorithmique et programmation :

- écriture de l'algorithme en pseudo code
- les affectations et l'utilisation de fonctions en Python

1. Ecrire un programme Scratch qui permet de calculer pour tout entier naturel  $n$  demandé en entrée à l'utilisateur la valeur de l'expression  $n^2 - (n + 1)^2 - (n + 2)^2 + (n + 3)^2$ .

Le faire fonctionner pour plusieurs valeurs différentes de  $n$  en entrée.

→ ***Beaucoup d'élèves ont des résultats différents pour les mêmes valeurs en entrée ...***

***La saisie des formules de calcul n'est pas très aisée avec Scratch !***

2. On propose de passer à un autre langage de programmation, on donne aux élèves un code Python de ce programme de calcul et on leur demande de le saisir et le faire tourner.

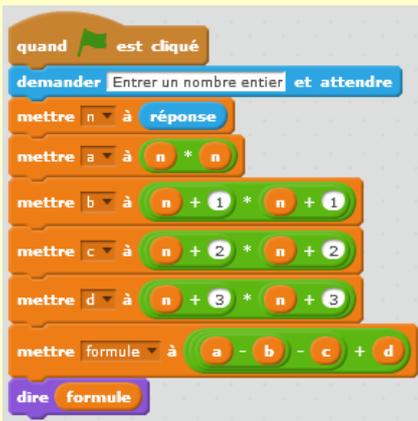
On fait le lien entre les deux en donnant le pseudo-code.

3. Faire plusieurs essais de calcul avec ce programme. Quelle conjecture peut-on faire ?
4. Démontrer cette conjecture.
5. Donner aux élèves le code Python de ce programme de calcul avec une fonction et leur demander de le saisir et le faire tourner. On constate qu'il fait la même chose.

On institutionnalise la notion de fonction en Python. On en profite pour leur montrer l'utilisation du mode console pour les fonctions définies dans le code.

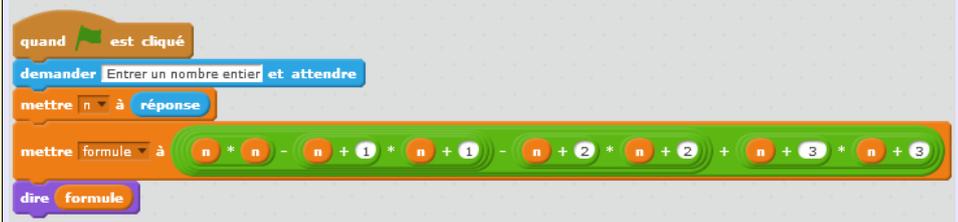
## Avec 4 affectations

### Scratch



## Avec 1 affectation

### Scratch



### Pseudo-code

```
a ← n2
b ← (n+1)2
c ← (n+2)2
d ← (n+3)2
formule ← a - b - c + d
```

### Pseudo-code

```
formule ← n2 - (n+1)2 - (n+2)2 + (n+3)2
```

### Python sans fonction

```
n=int(input("Saisir n "))
a=n**2
b=(n+1)**2
c=(n+2)**2
d=(n+3)**2
formule=a-b-c+d
print(formule)
```

### Python sans fonction

```
n=int(input("Saisir n "))
formule=n**2-(n+1)**2-(n+2)**2+(n+3)**2
print(formule)
```

### Python avec fonction

```
def formule(n):
    a=n**2
    b=(n+1)**2
    c=(n+2)**2
    d=(n+3)**2
    formule=a-b-c+d
    return formule
```

### Python avec fonction

```
def formule(n):
    resultat=n**2-(n+1)**2-(n+2)**2+(n+3)**2
    return resultat
```