

Cycle 4
5^e • 4^e • 3^e

Xavier Chanet

Algorithmique et programmation

avec Scratch



ellipses

Introduction et découverte de Scratch

Les problèmes que l'on rencontre habituellement en mathématiques trouvent en général leurs solutions dans le calcul à la main et le raisonnement. Il existe cependant toute une catégorie de problèmes pour lesquels ces techniques demeurent lourdes et inefficaces.

Pour comprendre, considérons les cinq problèmes suivants.

5 problèmes...

- Problème n°1** : Calculer $A = 2 + 5 \times 6$.
- Problème n°2** : Calculer $B = 25,68 \times 14,235 + 123,184$.
- Problème n°3** : Trouver le nombre x tel que $5x + 1 = 7$.
- Problème n°4** : Additionner tous les nombres entiers de trois chiffres.
- Problème n°5** : Quel est le plus petit nombre entier naturel n tel que $n^3 + 3n > 100\,000$?

Résolutions

Problème n°1 : On calcule aisément A , à la main.

$$\begin{aligned} A &= 2 + 5 \times 6 \\ &= 2 + 30 \quad (\text{la multiplication est prioritaire sur l'addition}) \\ &= 32 \end{aligned}$$

Problème n°2 : Le calcul de B nécessite une calculatrice pour ne pas perdre trop de temps. On obtient alors $B = 488,7388$.

Problème n°3 : Pour déterminer x , on soustrait d'abord 1 de chaque membre de l'égalité puis on les divise par 5.

$$\begin{aligned} 5x + 1 &= 7 \\ 5x + 1 - 1 &= 7 - 1 \quad (*) \\ 5x &= 6 \\ \frac{5x}{5} &= \frac{6}{5} \quad (**) \\ x &= \frac{6}{5} = 1,2 \end{aligned}$$

 Avec l'habitude on peut se passer d'écrire les égalités (*) et (**).

① *Les calculs à la main ou à la calculatrice et l'utilisation des règles algébriques de base nous ont permis de résoudre les trois premiers problèmes. Concernant la résolution des deux autres, l'usage de ces méthodes serait long, laborieux et source d'erreurs. Alors, comment les résoudre ?*

☑ **Problème n°4** : Les nombres entiers de trois chiffres sont 100, 101, 102, ..., 997, 998 et 999. Les additionner soi-même un à un prendrait trop de temps. Cependant, on peut indiquer la démarche à suivre à un ordinateur qui effectuerait alors lui-même les calculs quasi-instantanément. Concernant la méthode, on pourrait lui demander d'effectuer les actions :

- ▶ Partir de 0.
- ▶ Puis, ajouter 100, puis 101, puis 102, puis 103, ... ceci jusqu'à 999.
- ▶ Enfin, afficher le résultat.

Nous verrons dans le chapitre 1 comment traduire et faire exécuter ces actions par un ordinateur. Nous nous contenterons pour l'instant de dire qu'on obtiendra le résultat 494 550.

🔗 *Une série d'actions élémentaires, assemblées de manière logique, comme les précédentes, est appelée **algorithme**.*

Pour être interprété par un ordinateur, un algorithme devra être traduit au préalable dans un langage qu'il comprend, appelé langage de programmation. Il en existe des centaines, répondant chacun à des objectifs particuliers. Les plus connus sont Python, C, C++, C#, VB.net et Php. Dans ce cours, nous avons choisi de travailler avec le langage Scratch car c'est celui qui convient le mieux à l'apprentissage des notions abordées au collège.

☑ **Problème n°5** : On peut observer que l'inégalité $n^3 + 3n > 100\,000$ n'est pas vraie pour des petites valeurs de l'entier naturel n .

Par exemple, l'inégalité $n^3 + 3n > 100\,000$ est fausse pour $n = 0$:

- $0^3 + 3 \times 0 = 0$ et on n'a pas $0 > 100\,000$.

De même, elle est fausse pour $n = 1$ et $n = 2$:

- $1^3 + 3 \times 1 = 1 + 3 = 4$ et on n'a pas $4 > 100\,000$.
- $2^3 + 3 \times 2 = 8 + 6 = 14$ et on n'a pas $14 > 100\,000$.

Cependant, pour de grandes valeurs de n , l'inégalité $n^3 + 3n > 100\,000$ est vraie. Par exemple pour $n = 100$:

- $100^3 + 3 \times 100 = 1\,000\,300$ et on a bien $1\,000\,300 > 100\,000$.

Introduction et découverte de Scratch

De même que pour le problème n°4, il serait long et fastidieux de déterminer à la main la valeur de n cherchée.

On peut alors écrire un algorithme :

- ▶ On part de $n = 1$.
- ▶ Puis, tant que $n^3 + 3n$ ne dépasse pas 100 000, on augmente n de 1.
- ▶ Enfin on affiche la valeur de n obtenue.

Après avoir traduit cet algorithme puis exécuté le programme correspondant, on obtiendrait la valeur $n = 47$.

Présentation de Scratch

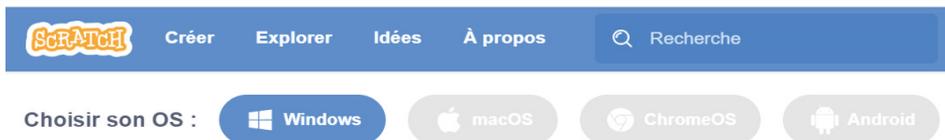
Scratch est un logiciel libre conçu dans le but de s'initier dès le plus jeune âge à l'algorithmique et à la programmation informatique.

Dans cet ouvrage, nous utilisons la version 3.29 de Scratch, qui est la plus récente au moment où il est écrit. Son approche est ludique et pratique pour apprendre sans difficultés de formalisme le codage informatique et les bonnes pratiques algorithmiques.

En effet, il n'est nul besoin de maîtriser une syntaxe d'écriture comme c'est traditionnellement le cas pour les langages de programmation usuels. Dans le cas de Scratch, il suffit de glisser, déposer et assembler des blocs pour définir les instructions à effectuer. Nous découvrirons plus en détail ce fonctionnement dans la prochaine partie.

Pour télécharger la version offline de Scratch 3.29 :

- Se rendre à l'adresse : <https://scratch.mit.edu/download>.
- Sélectionner son système d'exploitation (Mac OS, Windows ou Linux).
- Puis démarrer le téléchargement.



Introduction et découverte de Scratch

Après téléchargement puis installation, lancer Scratch. On arrive alors à l'écran ci-après (figure 0.2) où on peut distinguer :

- **Le menu des blocs d'instruction.**

Ces blocs sont des sortes de briques que l'on assemble pour créer des programmes. Ils sont classés par catégories.

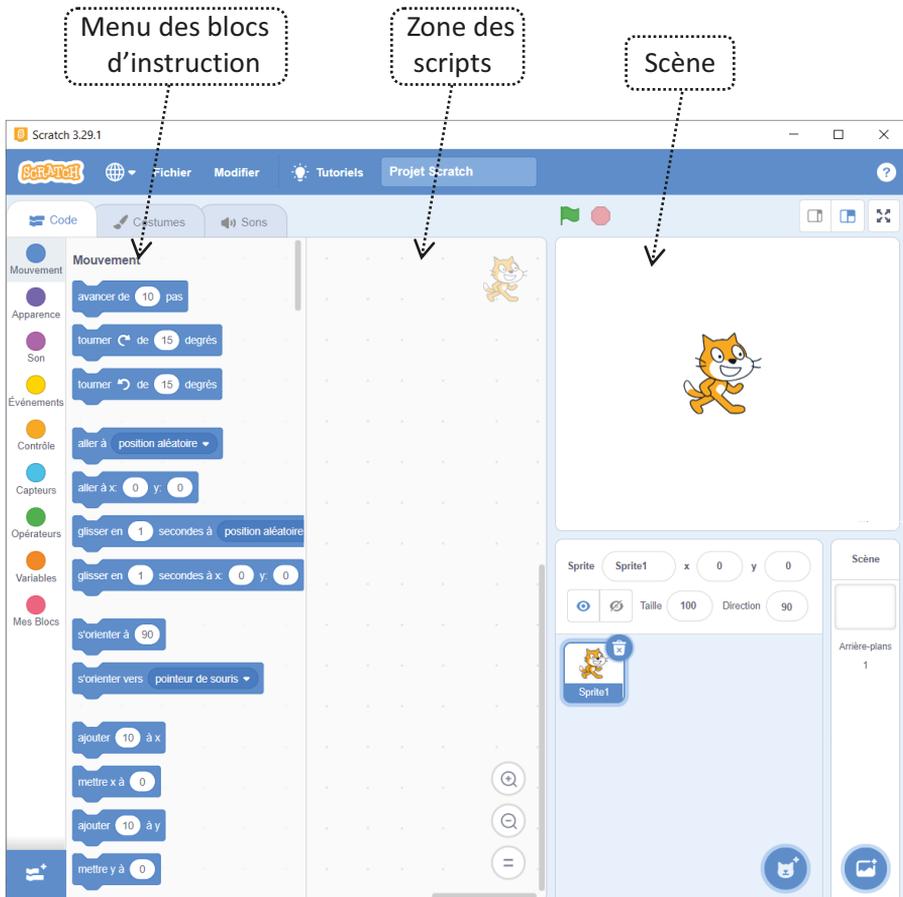
Les blocs « Mouvement » gèrent le mouvement des objets, les blocs « Apparence » gèrent leur apparence, etc.

- **La zone des scripts.**

C'est le lieu où on dépose et assemble les blocs par glisser-déposer.

- **La scène.**

C'est ici que ce déroule l'exécution du programme.



Outils algorithmiques

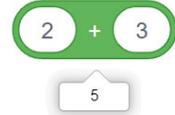
1.1 Données et opérations

🖥️ Activité de découverte ...

Lancer Scratch puis faire apparaître les blocs « Opérateurs ».

Le bloc  renvoie la somme de deux nombres.

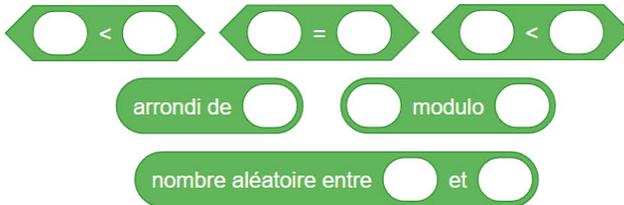
On peut le vérifier en réalisant quelques **tests** : par exemple, si on saisit 2 et 3 et si on clique ensuite sur le bloc, on obtiendra bien 5.



De même, les blocs suivants renvoient respectivement la différence, le produit et le quotient de deux nombres :



1) Réaliser des tests pour déterminer ce que retournent chacun des blocs suivants :



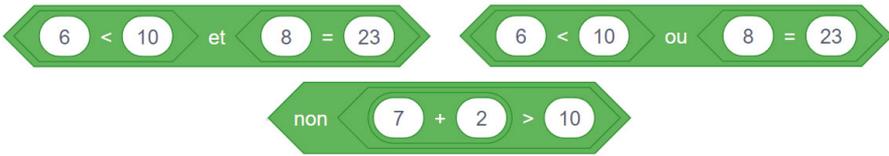
2) Insérer des caractères dans les blocs comme ci-dessous.



Réaliser d'autres tests avec ces blocs, puis définir ce que retourne chacun des blocs suivants :



3) Faire les assemblages suivants dans l'aire des scripts :



Cliquer sur chacun de ces trois blocs puis justifier les affichages obtenus.

4) Si on clique sur le bloc suivant, va-t-on obtenir vrai ou faux ? Justifier.



Vérifier ce résultat en réalisant l'assemblage, puis en cliquant dessus.

Corrigé

1) Le bloc renvoie vrai si le nombre de gauche est inférieur à celui de droite, faux sinon. Le bloc renvoie vrai si le nombre de gauche est supérieur à celui de droite, faux sinon. Le bloc renvoie vrai si les deux nombres saisis sont égaux, faux sinon.

Le bloc renvoie le reste de la division euclidienne du nombre de gauche par le nombre de droite. Le bloc renvoie l'arrondi à l'unité d'un nombre.

Le bloc renvoie un nombre au hasard compris entre le nombre de gauche et celui de droite. S'il s'agit de deux nombres entiers, un entier est renvoyé.

2) Le bloc renvoie la concaténation des deux chaînes de caractères saisies.

Par exemple, retourne *okazerty* :



Pour la saisie d'un nombre entier n et d'une chaîne de caractères, le bloc

retourne le $n^{\text{ème}}$ caractère de la chaîne.

Algorithmique et programmation

avec **Scratch**

Cycle 4
5^e • 4^e • 3^e

Ce livre s'adresse en premier lieu aux élèves de 5^e, 4^e et 3^e du collège et traite intégralement le thème « Algorithmique et programmation » du programme de mathématiques. Il constitue un véritable **outil d'accompagnement** permettant à chacun, quel que soit son niveau, de maîtriser toutes les connaissances et compétences attendues.

Après une **présentation** succincte du **logiciel Scratch**, on découvrira les **outils algorithmiques de base** : données, stockage, affichage et saisie, listes, instructions conditionnelles et boucles. Chaque nouvelle notion est abordée de manière expérimentale par le biais d'une **activité de découverte**, suivie d'un résumé de **ce qu'il faut retenir**, et enfin d'**exercices** pour aboutir à une bonne maîtrise des concepts. On pourra se perfectionner avec les **travaux pratiques** dont les thématiques sont très variées : sondage, dénombrement, calculs de dates, etc. L'accent sera aussi mis sur les déplacements des objets et leurs interactions, dans le but d'éditer des **applications ludiques**. On trouvera aussi des exercices issues des **Annales du brevet** des collèges.

Tous les exercices, travaux pratiques et autres activités sont **entièrement corrigés** de manière **détaillée** et **expliquée**. Les scripts Scratch apparaissent uniquement en couleur.

Cet ouvrage se veut aussi être une porte ouverte à une démarche de projet interdisciplinaire. Ainsi, à travers les nombreux exercices, on aura déjà l'occasion de revisiter des notions mathématiques familières telles que le calcul littéral, les fonctions, la géométrie et les probabilités. De plus, quelques exercices balayant un large champ de domaines (musique, analyse d'un texte, optimisation, jeux de hasard, etc.) peuvent déboucher sur de plus amples recherches individuelles ou en groupe.

Ce livre pourra également intéresser les enseignants, les parents et toute autre personne souhaitant s'initier à l'algorithmique et à la programmation.

www.editions-ellipses.fr

