

Sean McManus

**MON COURS
VISUEL DE
PROGRAMMI
MATION
SCRATCH ET PYTHON PAS À PAS**

Traduit de l'anglais (Royaume-Uni) par

Christine Eberhardt

DUNOD

L'édition originale de cet ouvrage a été publiée en 2020
au Royaume-Uni par Dorling Kindersley sous le titre
Beginner's Step-By-Step Coding Course

This edition published by arrangement
with Dorling Kindersley Limited,
a Penguin Random House Company.
All rights reserved.

Original Title : *Beginner's Step-by-Step Coding Course*
Copyright © Dorling Kindersley Limited, 2020
DK, One Embassy Gardens, 8 Viaduct Gardens
London SW11 7BW
A Penguin Random House Company

Pour l'édition française :
© Dunod, 2021
11 rue Paul Bert, 92240 Malakoff
www.dunod.com
978-2-10-081936-2

Traduction : Christine Eberhardt

Interdit à la vente au Québec

Code de la propriété intellectuelle n'autorisant, aux termes de l'article
122-5, 2° et 3° a), d'une part, que les « copies ou reproductions strictement
servées à l'usage privé du copiste et non destinées à une utilisation collective »
d'autre part, que les analyses et les courtes citations dans un but d'exemple et
d'illustration, « toute représentation ou reproduction intégrale ou partielle faite
sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est
illicite » (art. L. 122-4).

La représentation ou reproduction, par quelque procédé que ce soit, constitue
sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, une
contrefaçon sanctionnée par les articles L. 335-2 et suivants du
Code de la propriété intellectuelle.

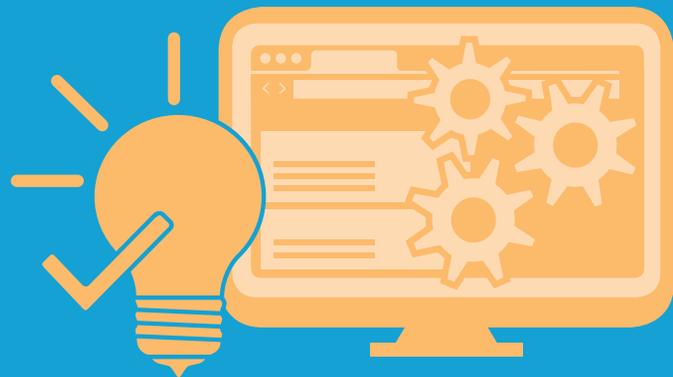
SOMMAIRE

À propos de ce livre 6

INTRODUCTION

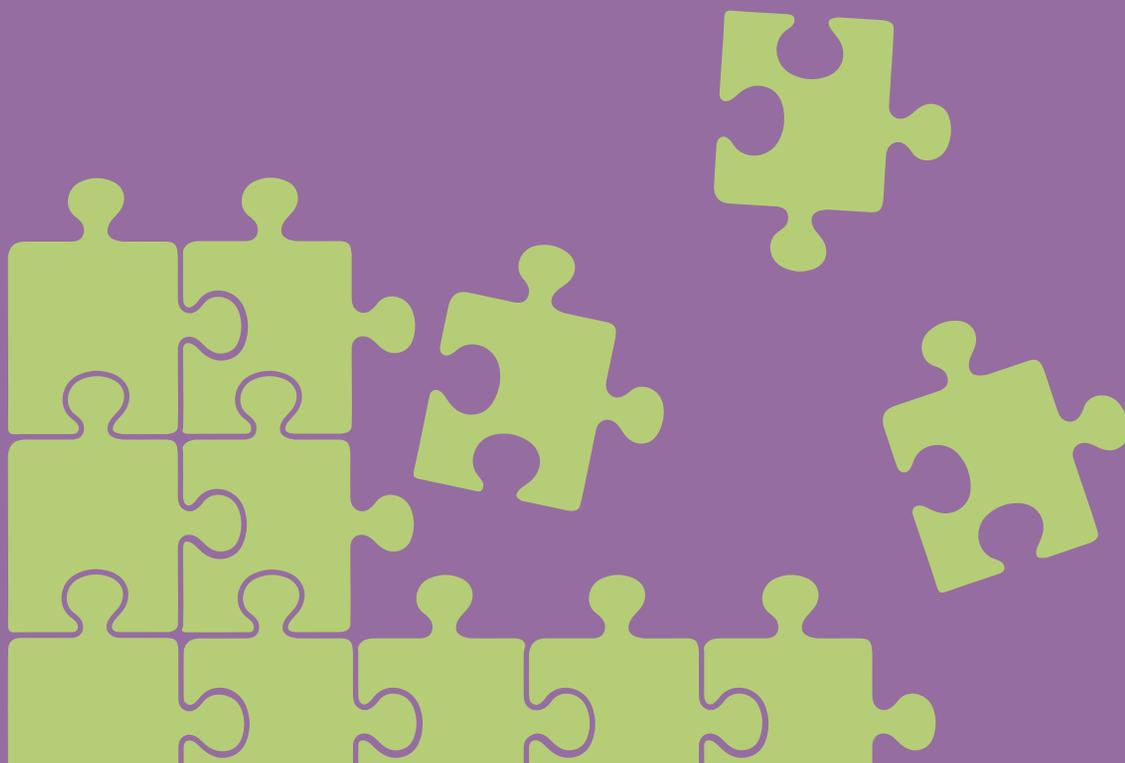
Qu'est-ce que la programmation? 8

Les langages de programmation 10



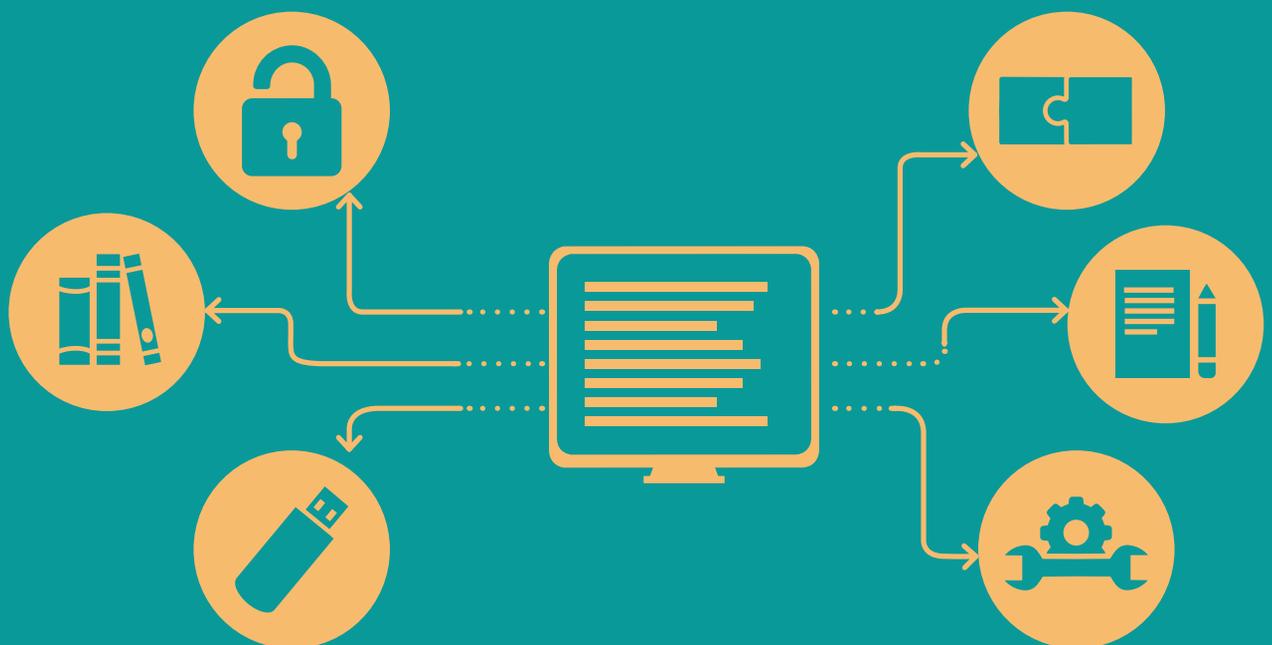
SCRATCH

Qu'est-ce que Scratch?	16	Logique et décisions	32
L'interface de Scratch	18	Les entrées	34
Les sprites	20	Envoyer des messages	36
Blocs de couleur et scripts	22	Utiliser les fonctions	38
Le déplacement des sprites	24	Traducteur de voyage	40
Apparences et sons	26	Casse-tête	52
Gérer le déroulement du programme	28	Esquive d'astéroïdes	68
Manipuler les données	30		



PYTHON®

Qu'est-ce que Python?	82	Les bibliothèques	104
Installation de Python	84	Générateur d'équipes	106
Utilisation de IDLE	86	Débogage	118
Les variables dans Python	88	Planificateur de projet	122
Les données dans Python	90	Objets et classes	144
Opérateurs logiques et branchements	92	Gestionnaire de budget	146
Entrée et sortie	94	Pygame Zero	164
Les boucles dans Python	96	Knight's Quest	166
Les fonctions	100		



À propos de ce livre

Organisation de l'ouvrage

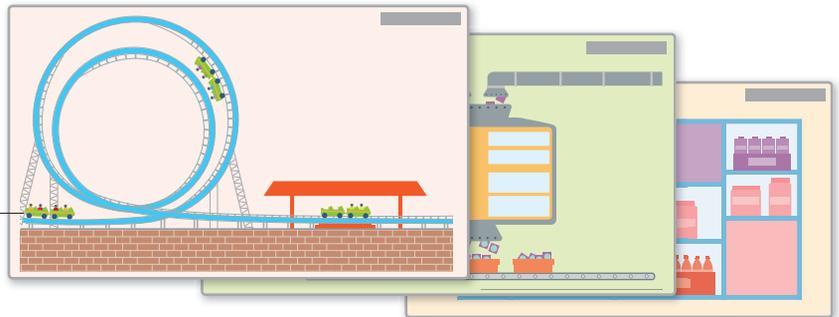
Deux chapitres exposent les principes fondamentaux des deux langages de programmation Scratch et Python. Pour chaque langage les principes de base

sont présentés puis des projets vous sont proposés afin que vous puissiez tester par vous-même les notions abordées.

Concepts

Chaque chapitre présente les concepts de programmation de base du langage à l'aide d'exemples de code pratiques que vous pouvez tester pour mieux comprendre le concept.

Les illustrations vous aident à comprendre les concepts



Projets

Les projets de cet ouvrage vont vous apprendre à créer des jeux, des planificateurs et des applications. Chacun d'eux débute par un rapide aperçu de ce que vous allez apprendre, de comment le planifier et de quoi vous aurez besoin pour le créer. Des instructions simples étape-par-étape vous guident en expliquant chaque caractéristique du code, à l'aide d'annotations détaillées.

Cet encadré présente les concepts utilisés dans le projet

VOUS ALLEZ APPRENDRE

■ ■ ■ ■ ■

Durée: 1 heure

Lignes de code: 58

Niveau de difficulté

● ● ● ● ●

Indique le temps estimé pour réaliser un projet

Indique le nombre de lignes de code estimé pour un projet. Ce nombre varie selon l'éditeur de code utilisé

Indique le niveau de difficulté d'un projet, 1 étant le plus facile

VOUS ALLEZ APPRENDRE

Les projets sont découpés en sections avec des étapes bien définies afin de faciliter l'apprentissage

1.1

■ ■ ■ ■ ■

1.2

■ ■ ■ ■ ■

disponible = 2500.00

budgets = {}

ÉTAPE PAR ÉTAPE

Fenêtres de code Python

Le code Python est édité dans deux fenêtres différentes; la fenêtre de commande et l'éditeur. Dans cet ouvrage, des couleurs différencient ces fenêtres à grille afin que vous sachiez dans laquelle vous devez saisir le code.

Chaque rectangle de la grille représente un espace dans le code

```
>>> input = 2
>>> score = input * 3
>>> print(score)
6
```

FENÊTRE DE COMMANDE

Ces chevrons n'apparaissent que dans la fenêtre de commande. Saisissez le code après l'invite >>>

```
def reset_jeu():
    global score, charmes
    score = 0
    charmes = 0
```

FENÊTRE DE L'ÉDITEUR

Chaque indentation (espace au début d'une ligne) représente quatre cases de grille vides. Les indentations supplémentaires sont toujours en multiples de quatre cases

Icône « Enregistrer »

L'icône « Enregistrer » vous rappelle à quel moment vous devez sauvegarder le programme au cours du projet.



ENREGISTRER

Trucs et astuces

La section Trucs et astuces en fin de projet donne quelques astuces pour améliorer le code existant ou ajouter de nouvelles fonctionnalités.



Code sur le site [dunod.com](http://www.dunod.com)

Le pack de ressources des projets de cet ouvrage (à l'exception des sections « Trucs et astuces » et des projets créés avec Scratch) est disponible sur www.dunod.com. Vous y trouverez le code au format original (.py) et les images pour tous les jeux.



Rendez-vous à cette adresse pour télécharger le pack de ressources de cet ouvrage

Qu'est-ce que la programmation ?

Les ordinateurs et appareils électroniques ont besoin de logiciels (ou programmes) pour fonctionner. La programmation, ou codage, est l'art d'écrire des instructions. Il existe des programmeurs professionnels mais le codage peut aussi être un passe-temps.

Les programmes informatiques sont partout

La programmation ne concerne plus uniquement les systèmes informatiques conventionnels. Le monde est devenu de plus en plus numérique et presque tout fonctionne avec des logiciels. Les programmes sont désormais intégrés dans des appareils tels que les téléphones portables et les tablettes, les équipements domestiques et même dans les systèmes de transport.

DEVENIR PROGRAMMEUR

Une fois que vous maîtrisez les bases, voici comment développer vos compétences en développement.

- **Pratiquez** : écrivez et testez du code
- **Lisez du code** : vous apprendrez beaucoup en étudiant les programmes des autres
- **Apprenez plusieurs langages** : en étudiant les concepts sur lesquels reposent les autres langages, vous saurez lequel choisir pour votre projet
- **Publiez des projets** : en publiant votre travail, vous bénéficierez des commentaires des autres programmeurs et vous améliorerez votre code

Centre de données

Un centre de données est similaire à une installation informatique à l'échelle industrielle. Ses nombreux serveurs accessibles sur Internet stockent des données ou exécutent un logiciel « dans le cloud ». Le « cloud » est un réseau mondial de serveurs qui fournissent des services via Internet.

Voiture

Un logiciel peut être utilisé pour contrôler la vitesse, la température ou la consommation d'une voiture. Le système de navigation nécessite également des programmes.

Ordinateur de bureau

Les ordinateurs sont largement utilisés dans les bureaux et les foyers pour gérer les activités administratives et créatives comme la musique, le design, le traitement de texte, les comptes, etc.

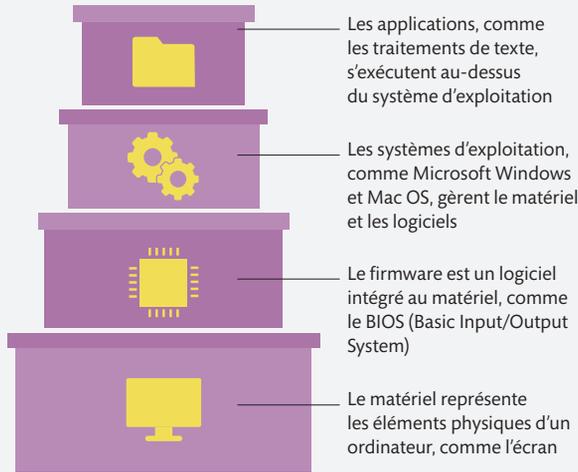
Machine à laver

De nombreux appareils ménagers sont équipés de programmes. Les cycles de lavage, la température de l'eau et les programmations d'une machine à laver, par exemple, sont gérés par un logiciel.



Qu'est-ce qu'un programme ?

Un programme est un ensemble d'instructions qu'un ordinateur suit pour effectuer une tâche. Les programmes peuvent être extrêmement complexes. Microsoft Windows, par exemple, se compose de millions de lignes d'instructions.

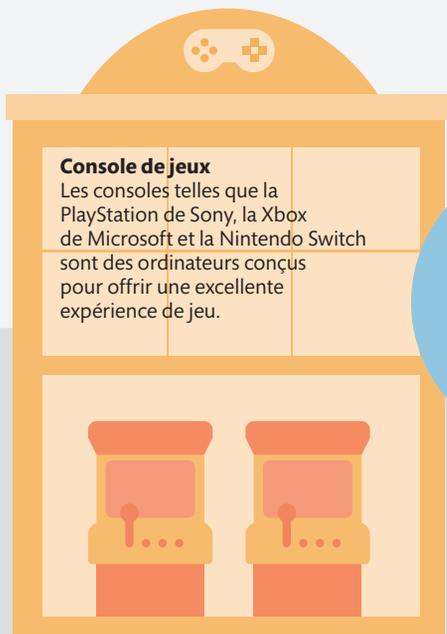


Penser comme l'ordinateur

Pour écrire un programme, il faut comprendre comment l'ordinateur traite les instructions. Chaque tâche doit être décomposée. On ne peut pas simplement demander à un robot de « faire griller des toasts », par exemple. Il faut programmer des instructions précises et détaillées pour chaque étape du processus.

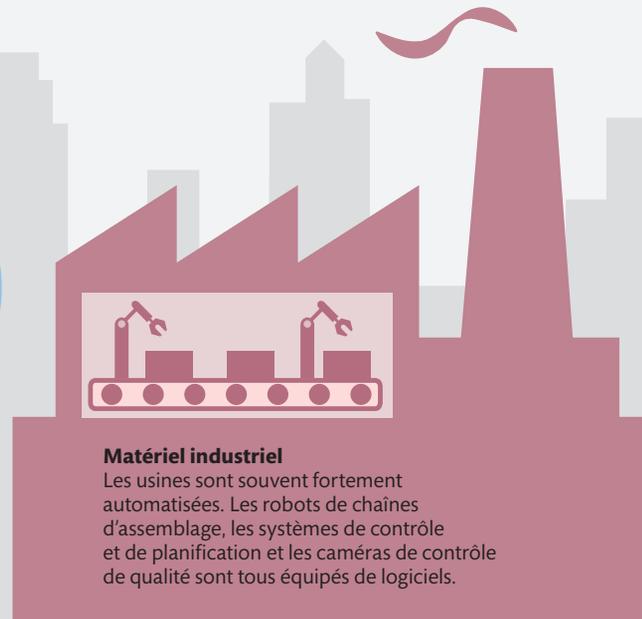
1. Ouvrir la corbeille à pain
2. Prendre le pain
3. Ouvrir le sac
4. Prendre une tranche
5. L'insérer dans le grille-pain
6. Prendre une tranche
7. L'insérer dans le grille-pain
8. Enfoncer le bouton
9. Attendre que les toasts sautent

Au lieu de répéter deux fois la même instruction, il sera plus rapide et plus simple de dire dans le programme : « Répéter deux fois : prendre la tranche, l'insérer dans le grille-pain »



Appareil photo

Sur les appareils photo d'aujourd'hui, c'est un logiciel qui permet de changer les réglages, de revoir les photos et de les supprimer au besoin.



Les langages de programmation

Un langage de programmation est un ensemble de mots et de symboles qui permet d'écrire des instructions pour un ordinateur. Il faut parfois choisir entre la facilité d'utilisation du langage et sa puissance.

Langages de haut et de bas niveaux

Les langages de programmation de haut niveau n'exigent pas de connaître le matériel informatique. Ils emploient souvent des mots proches du langage humain et gèrent automatiquement certaines fonctions de l'ordinateur. Souvent, un programme écrit dans ce

langage peut s'exécuter sur différents matériels. Les langages de bas niveau offrent un contrôle plus pointu de l'ordinateur, mais ils exigent une compréhension plus approfondie de son fonctionnement. Les programmes écrits dans ces langages peuvent ne pas fonctionner sur d'autres matériels.

LANGAGES DE HAUT NIVEAU

- Écriture assez rapide
- Facile à comprendre
- Rapidité d'exécution acceptable
- Programmes compatibles avec différents matériels
- Aucune connaissance informatique requise

Instruction pour afficher du texte

```
print("Bonjour!")
```

Python

Langage populaire de haut niveau, Python est facile à lire et à écrire. Cette instruction affiche « Bonjour! » sur l'écran.



LANGAGES DE BAS NIVEAU

- Contrôle direct des fonctionnalités matérielles
- Vitesse d'exécution supérieure avec du code bien écrit
- Nécessite une bonne connaissance du matériel
- Programme uniquement compatible avec le même processeur ou un processeur similaire

Nombre hexadécimal, un système d'écriture en base 16 courant en informatique

```
MOV AX, 66H
```

Assembleur

Ce langage est très proche du code machine du processeur. Cet exemple prend un nombre et le place dans l'accumulateur du processeur.



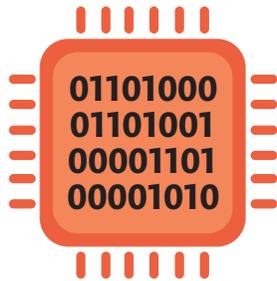


Code machine

Le code de bas niveau qui est directement reconnu par le matériel informatique et le processeur est appelé code machine, ou langage machine. C'est une collection de chiffres binaires – 0 et 1 – que le processeur lit et interprète. Les instructions en code machine se composent d'un nombre appelé *opcode*, ou code opération, qui détermine la nature de l'instruction, et d'un ou plusieurs *opérandes*, qui indiquent quelles données utiliser.

Microprocesseur

Le microprocesseur est le «cerveau» de l'ordinateur: il contrôle la plupart des opérations. Il exécute les commandes et les instructions de code machine.



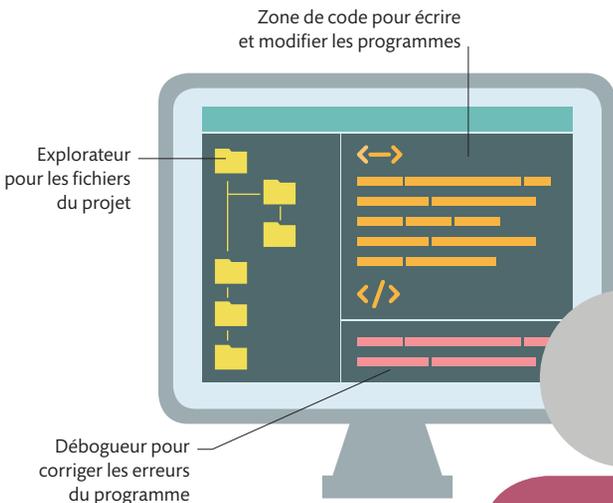
Comment l'ordinateur comprend un langage de programmation

Tous les programmes finissent sous forme de code machine. La plupart sont écrits dans des langages conviviaux qui doivent être traduits en bits pour pouvoir être exécutés par un processeur. Un *interpréteur* traduit et exécute les instructions pendant l'exécution du programme, alors que les *compilateurs* traduisent le programme une fois pour toutes avant qu'il ne s'exécute.



Utilisation d'un environnement de développement intégré (EDI)

Un EDI est un ensemble d'outils pour les programmeurs. Il contient un éditeur de code et fournit souvent des fonctions telles que la saisie semi-automatique et l'affichage en couleurs pour faciliter sa lecture. Certains EDI proposent aussi un débogueur et un compilateur ou un interpréteur pour tester et exécuter les programmes.



APPLICATIONS

Une fois que vous avez appris à programmer, ces compétences peuvent être utilisées pour un large éventail de projets créatifs.

- **Domotique:** pour contrôler certaines choses à distance comme l'éclairage ou les volets roulants
- **Jeux:** c'est un excellent moyen d'expérimenter le codage car ils sont faciles à partager et on peut donc obtenir un bon retour dessus (voir pages 68-79, 166-191)
- **Robots:** grâce aux cartes Arduino ou Raspberry Pi combinées à des kits ou composants électroniques, on peut programmer ses propres robots.
- **Sites web et applications web:** apprendre à coder en HTML, CSS ou JavaScript permet aussi de créer des programmes exécutables directement depuis un navigateur Internet

Exemple d'interface

Avec certains IDE, vous pouvez configurer votre environnement de travail. Ici le programmeur parcourt les fichiers du projet à gauche, écrit et modifie le code à droite et contrôle ses erreurs en bas.

Types de langages de programmation

De nombreuses philosophies ou paradigmes ont été utilisés pour concevoir des langages de programmation. N'étant pas mutuellement exclusifs, ils incarnent souvent différentes notions fondamentales. Ils peuvent aussi être utilisés en fonction de l'approche choisie par le programmeur.

Python, par exemple, peut être utilisé en programmation orientée objet ou procédurale. JavaScript peut être utilisé en programmation événementielle ou orientée objet. L'approche et le langage choisis sont souvent une question de préférence du programmeur. Voici quelques définitions et classements des langages de programmation.

PROGRAMMATION IMPÉRATIVE

Ces langages se traduisent par une liste d'instructions à exécuter par l'ordinateur. Le programmeur doit d'abord déterminer comment la tâche peut être effectuée, puis fournir les instructions étape par étape. Les langages impératifs sont courants et incluent Python (voir page 80), C, C++ et Java.

```
user = input("Quel est votre nom?")  
print("Hello", user)
```

SAISIE DANS PYTHON

Programme python qui souhaite la bienvenue à l'utilisateur

```
Quel est votre nom? Marius  
Hello Marius
```

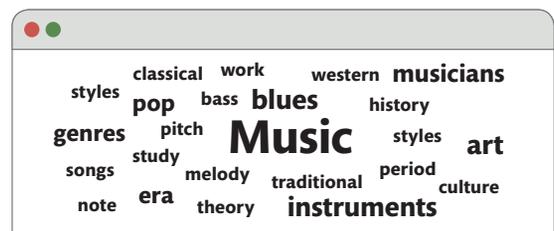
SORTIE DE PYTHON

PROGRAMMATION DÉCLARATIVE

En programmation déclarative, les programmeurs indiquent à l'ordinateur le résultat souhaité, sans avoir besoin de préciser comment il sera atteint. Avec Wolfram, par exemple, un nuage de mots basé sur les mots de la page Musique de Wikipédia peut être créé en utilisant une seule ligne. D'autres langages déclaratifs incluent SQL, qui est utilisé pour les bases de données.

```
WordCloud[WikipediaData["music"]]
```

SAISIE DANS WOLFRAM



SORTIE DE WOLFRAM

PROGRAMMATION ÉVÉNEMENTIELLE

En programmation événementielle, le programme surveille l'apparition d'un événement puis démarre la séquence d'instructions appropriée quand il se produit. Par exemple, un programme peut réagir aux actions de l'utilisateur, à l'entrée d'un capteur ou aux messages d'autres systèmes informatiques. JavaScript et Scratch (voir page 14), entre autres, peuvent être utilisés pour écrire des programmes orientés événements.

Crée un bouton de page web

```
<input type="button" value="Cliquez ici!"  
onClick="showMessage();" >
```

Un clic sur le bouton déclenche l'exécution des instructions JavaScript de `showMessage()`

Cette icône signale que la ligne de code a été coupée et continue sur une seconde ligne



CHOIX DU LANGAGE

Parfois, les programmeurs choisissent leur langage en fonction du matériel, de l'équipe avec laquelle ils programment ou du type d'application qu'ils souhaitent créer. Ils ont souvent le choix.

PYTHON

Langage flexible dont le code est facile à comprendre

JAVA

Largement utilisé dans la finance, les petits appareils et téléphones Android

SCRATCH

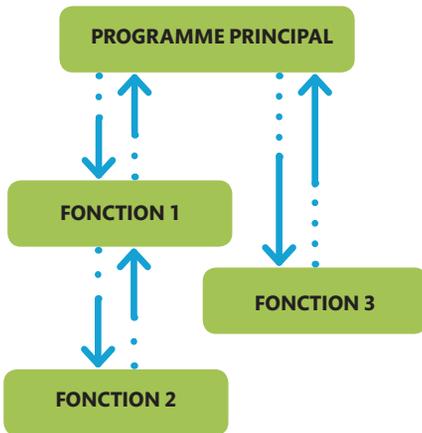
Parfait pour débiter, Scratch est idéal pour les jeux simples.

JAVASCRIPT

Utilisé pour l'interactivité des pages web.

PROGRAMMATION PROCÉDURALE

Programmation basée sur des fonctions, qui constituent des éléments de programme réutilisables. Les fonctions peuvent exécuter d'autres fonctions et même se relancer. Elles simplifient le développement, le test et la gestion des programmes. De nombreux langages tels que Java et Python (voir page 80) prennent en charge la programmation procédurale.



PROGRAMMATION ORIENTÉE OBJET

En programmation orientée objet, les données et les instructions associées sont stockées dans des « objets ». Ceux-ci peuvent interagir pour répondre aux besoins du programme. L'objectif est de rendre le code plus modulaire, donc plus facile à gérer et réutilisable. De nombreux langages de programmation courants, tels que C++, JavaScript et Python, prennent en charge la programmation orientée objet.

OBJET

Données pour cet objet

Instructions pour cet objet

Interfaces de communication

LANGAGES DE PROGRAMMATION VISUELS

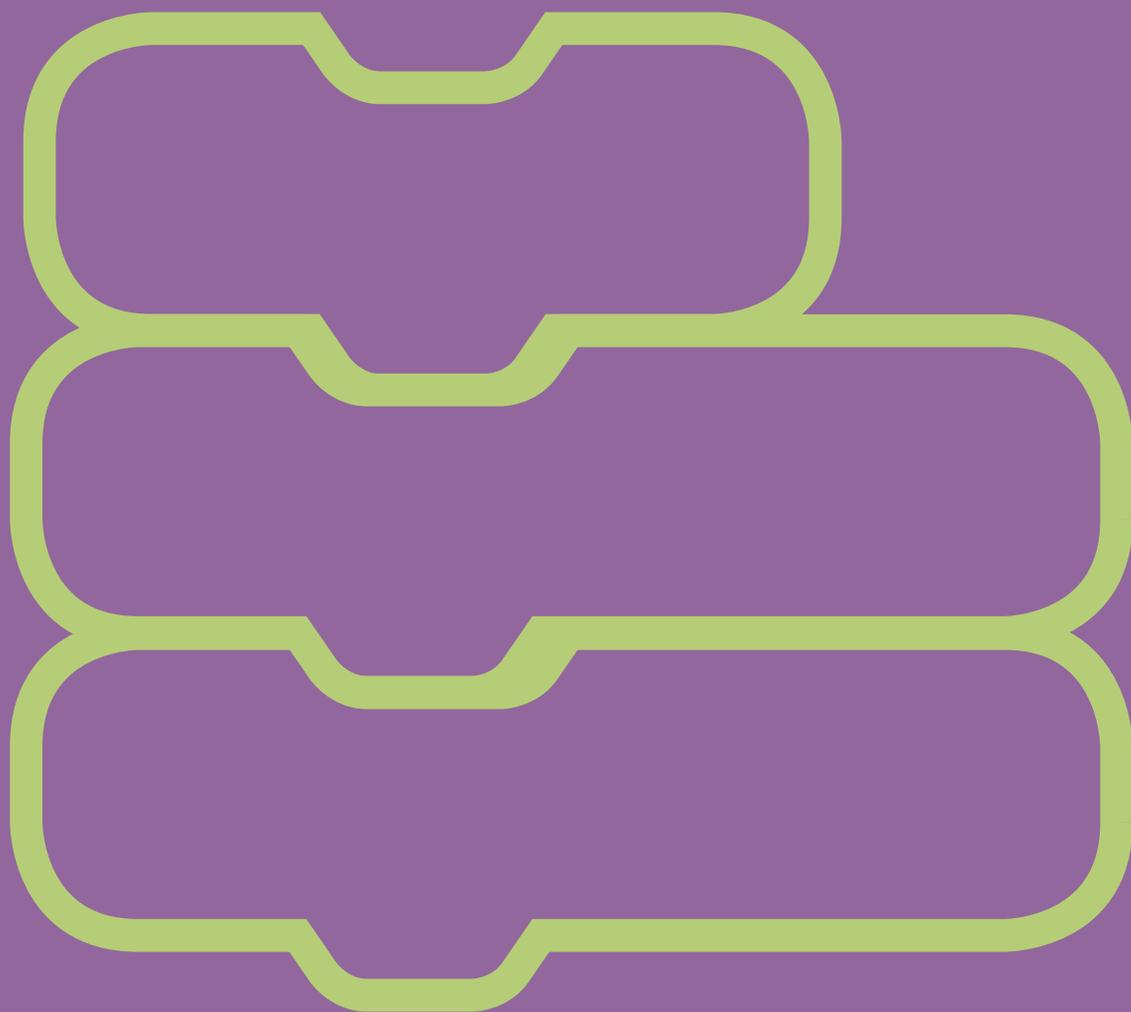
Ces langages facilitent le développement grâce à leurs interfaces basées sur le glisser-déposer. La programmation s'effectue plus rapidement et avec moins d'erreurs. Visual Basic, par exemple, propose des outils pour concevoir visuellement des interfaces utilisateur. Scratch (voir page 14) est un autre langage très visuel, souvent utilisé pour apprendre la programmation.

Programme Scratch qui réagit lorsqu'on clique sur un bouton

Quand ce sprite est cliqué

dire **Le bouton a été cliqué!** pendant **2** secondes





SCRATCH

Qu'est-ce que Scratch ?

Scratch est un langage de programmation visuel qui n'implique aucune saisie de code. Les programmes sont construits à l'aide de blocs colorés qui représentent des instructions. Scratch met l'accent sur l'aspect créatif du codage et permet aux utilisateurs de créer des jeux interactifs, des histoires et d'autres applications visuelles.

Fonctionnalités

Les fonctionnalités de Scratch en font un langage idéal pour les débutants.

L'utilisation de blocs de code prêts à l'emploi, en particulier, le distingue de la plupart des autres langages.

Glissé-déposé

Les blocs de Scratch sont glissés et déposés dans la zone de code pour créer des programmes. Il y a très peu de frappe, les utilisateurs sont donc moins susceptibles de faire des erreurs.

Communauté

Les utilisateurs de Scratch peuvent communiquer entre eux via une communauté Scratch intégrée. Ils peuvent partager leurs programmes pour que d'autres puissent jouer avec, les modifier ou les remixer. Ils peuvent également apprendre en étudiant les projets des autres.

Langage puissant

Scratch est facile à utiliser, mais applique des concepts de base identiques à ceux des langages professionnels. Il représente donc une bonne introduction à la programmation.

Conception en puzzle

Les blocs d'instructions s'emboîtent comme des pièces de puzzle, il n'est donc pas possible de créer de mauvaises combinaisons. Les erreurs de logique sont ainsi minimisées.



Apprendre à programmer avec Scratch

Scratch a été créé par le groupe Lifelong Kindergarten du Massachusetts Institute of Technology (MIT) aux États-Unis en 2007.

Scratch est ludique et très facile à utiliser pour débuter, c'est pourquoi il est largement utilisé dans l'éducation. Son interface très visuelle se compose

de blocs de code colorés qui s'assemblent pour former des scripts. Ces derniers peuvent intégrer des images et du son pour créer une action à l'écran. Scratch fournit une plateforme très performante (voir pages 18-19) pour l'apprentissage de la programmation.

Ressources intégrées

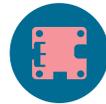
Scratch est livré avec une bibliothèque préinstallée de sons et d'images (appelés *sprites*, voir pages 20-21). Les autres langages n'offrent pas une telle facilité, il faut souvent créer ou télécharger les images avant d'écrire un programme.

Codage par couleurs

Chaque bloc d'instruction pour le mouvement, le son, le contrôle et la détection (entre autres) possède une couleur facilement identifiable lors de la création d'un programme.

Support matériel

La dernière version de Scratch fonctionne avec Windows, macOS, Linux et même sur tablette. Des extensions fournissent l'interaction avec des périphériques matériels.



Raspberry Pi

Scratch peut utiliser un Raspberry Pi pour se connecter à d'autres capteurs ou moteurs.



micro:bit

Scratch peut être utilisé avec un BBC micro:bit doté d'un écran LED intégré, de boutons et de capteurs d'inclinaison.



Lego®

Scratch peut se connecter à Lego® WeDo et Lego® Mindstorms™ pour contrôler des moteurs, capteurs et robots.



Webcam

Scratch est capable d'accéder à une webcam pour superposer des images à un flux vidéo afin de créer des applications en réalité augmentée.

OÙ TROUVER SCRATCH

L'environnement de développement Scratch est requis pour utiliser Scratch. Il est accessible en ligne et hors ligne.



Dans votre navigateur

Rendez-vous sur site Web Scratch <https://scratch.mit.edu/> et cliquez sur Rejoindre pour créer un compte.



Hors ligne

Vous pouvez télécharger Scratch pour travailler hors ligne depuis le site <https://scratch.mit.edu/download>.

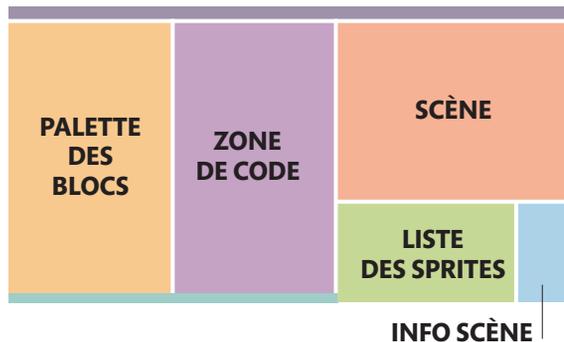
L'interface de Scratch

C'est dans l'interface de Scratch que vous créez des programmes, que vous les éditez et que vous en affichez le résultat. L'interface est divisée en plusieurs sections, chacune ayant un objectif particulier. Ce livre utilise Scratch 3.0 – la dernière version de Scratch.

Comprendre l'interface

L'interface de Scratch se compose des sections suivantes :

- **Palette des blocs** : elle contient les instructions, ou blocs de code, nécessaires pour construire un programme
- **Zone du code** : c'est là que les blocs de code sont assemblés pour créer un script
- **Scène** : zone d'interaction de l'utilisateur avec le programme
- **Liste des sprites** : affiche et gère toutes les images, ou sprites, du programme
- **Info scène** : gère les images d'arrière-plan



VERSIONS DE SCRATCH

Scratch existe actuellement en trois versions, chacune avec une disposition d'écran différente. Les fonctionnalités et blocs d'instructions les plus récents ne fonctionnent pas tous avec les versions antérieures.

- **Scratch 1.4** : Interface similaire à celle de Scratch 3.0, la zone de code s'appelait l'espace des scripts
- **Scratch 2.0** : La scène se trouvait à gauche. Introduction du clonage des sprites et réorganisation de certains blocs d'événements.
- **Scratch 3.0** : Introduction des extensions et déplacement des blocs stylo dans ces derniers.

Création d'un nouveau projet, chargement depuis ou enregistrement sur l'ordinateur

Choix de la langue

Modification des projets existants

Cliquez sur l'onglet Son pour ajouter des effets sonores ou de la musique aux sprites

SCRATCH

Fichier Modifier Tutoriels

Code Costumes Sons

Mouvement

Apparence

Son

Événements

Contrôle

Capteurs

Opérateurs

Variables

Mes blocs

Cliquez sur l'onglet Costumes pour changer l'apparence du sprite

avancer de 10 pas

tourner 15 degrés

tourner 15 degrés

aller à position aléatoire

aller à x: 0 y: 0

glisser en 1 secondes à position aléatoire

glisser en 1 secondes à x: 0 y: 0

s'orienter à 90

s'orienter vers pointeur de souris

Faites glisser les blocs d'instructions dans la zone de code et emboîtez-les pour construire un programme

Contient les blocs d'instruction disponibles

Sac à dos

Utilisez le bouton «Ajouter une extension» pour ajouter de nouvelles catégories de blocs à la palette des blocs

Stockez le code, les sprites, les costumes et les sons intéressants dans le sac à dos pour les réutiliser dans d'autres projets