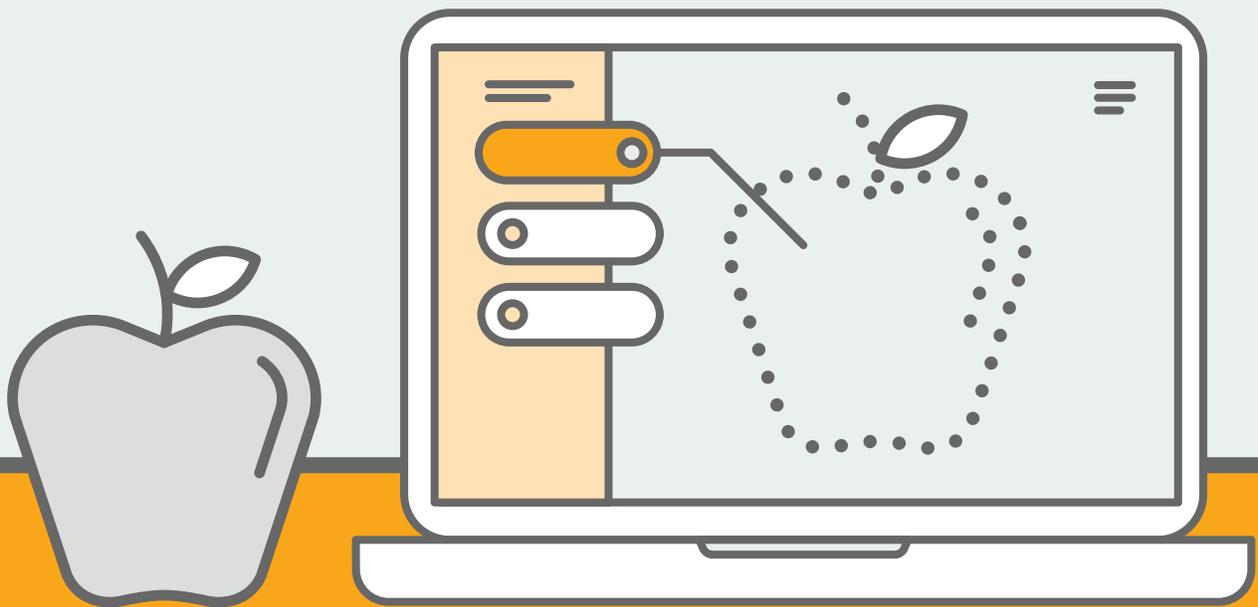


✘ enseignants en  
programmation



# COFFRE À OUTILS NUMÉRIQUE :

Guide de démarrage



Un programme éducatif de  
**CANADA EN PROGRAMMATION**

# Table des matières

## INTRODUCTION D'ENSEIGNANTS EN PROGRAMMATION

- 4 Qu'est-ce que Teachers Learning Code?
- 5 La création de Teachers Learning Code
- 6 Notre formule pour bien enseigner la programmation
- 6 Le coffre à outils numérique d'Enseignants en programmation

## ENSEIGNER LA PROGRAMMATION POUR LES DÉBUTANTS

- 7 Pourquoi enseigner la programmation?
- 8 Question de diversité
- 9 Objectifs d'apprentissage
- 10 Pleins feux sur la pensée informatique
- 10 Glossaire de la programmation

**Pour plus de plans de cours, visitez:**  
<https://www.canadalearningcode.ca/fr/plans-de-cours/>

## PLANIFICATION DES COURS

- 14 Exigences minimales
- 15 Horaire de préparation
- 16 Donner le ton
- 17 Structure de cours suggérée
- 17 Rythme
- 17 Gestion des imprévus

## EXTRAS

30 Notes

# Bienvenue

NOUS SOUHAITONS INSPIRER LES CANADIENS 

en leur enseignant la programmation afin de leur permettre de devenir des créateurs et citoyens numériques. Ainsi, ils seront bien outillés pour comprendre notre nation et le monde entier. Ils pourront se servir de ces outils pour participer à la société et l'influencer en tant que créateurs et innovateurs de la technologie.

Que vous soyez un enseignant dans une classe, un coordonnateur de programme dans un centre communautaire, un parent faisant l'école à la maison ou un guide chez les scouts, nous avons conçu ce guide pour vous aider à enseigner la programmation à tout le monde.

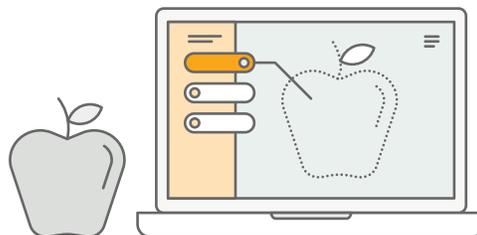
Grâce à nos nombreuses années d'expérience, nous avons appris quelques trucs pour enseigner la programmation. Dans ces guides D'Enseignants, nous partagerons avec vous quelques-uns de nos conseils pour commencer, des ressources pour vous initier à la programmation, ainsi que des activités qui sont aussi faciles à comprendre qu'à organiser pour enseigner la programmation à la prochaine génération de technologues du Canada.

Nous vous encourageons à utiliser nos ressources à votre guise, que ce soit en modifiant les leçons ou en en créant de nouvelles que vous pourrez partager avec notre communauté. Nos guides ne sont qu'un point de départ pour démarrer votre programme : le parcours et la destination sont à votre entière discrétion!

**La technologie se prête à la création. Amusez-vous!**

## INTRODUCTION

# enseignants en programmation



## Qu'est-ce que d'Enseignants en programmation?

L'objectif : utiliser la technologie pour changer le monde grâce au travail d'équipe, à la créativité, et, bien sûr, au code.

Enseignants en programmation est un programme conçu par Canada en programmation, un organisme de bienfaisance œuvrant à l'échelle nationale. Son but est d'inspirer les jeunes à devenir des créateurs de la technologie plutôt que de simples consommateurs en suivant des cours et en réalisant des exercices de programmation.

Enseignants en programmation enseigne une méthode d'enseignement de la programmation dont le contenu peut être simplifié ou enrichi selon les besoins des apprenants.

Tous les organisateurs de programme jeunesse ainsi que les enseignants, qu'ils aient une formation technique ou non, sont en mesure de donner nos cours dans les écoles et les centres communautaires.

Le contenu de nos guides pratiques et nos exercices de programmation ont été pensés pour les élèves du primaire et du secondaire. Ils ont été conçus avec les mêmes éléments éprouvés qui sont à l'origine du succès fulgurant de nos programmes.

> [enseignantsenprogrammation.com](https://enseignantsenprogrammation.com)

## Le développement de d'Enseignants en programmation

Canada en programmation a enseigné à des milliers de jeunes d'un océan à l'autre dans le cadre de ses ateliers, camps et programmes parascolaires Girls Learning Code, Kids Learning Code et Teens Learning Code. Notre contenu a été élaboré en partenariat avec des éducateurs et des experts du domaine. Ensuite, nous l'avons mis à l'essai de nombreuses fois avec des jeunes de trois à dix-sept ans tant à l'intérieur et qu'à l'extérieur de la classe.

Pour le programme d'Enseignants en programmation, nous avons peaufiné la méthodologie et le processus de conception du contenu. En effet, nous avons fait appel à des éducateurs n'ayant pas de connaissances techniques pour concevoir, élaborer et mettre en œuvre le matériel. Dans le but de concevoir un programme adapté tant aux enseignants qu'aux apprenants, nous avons formé des groupes de clientèle, envoyé des sondages, enseigné aux côtés d'enseignants dans des écoles et des centres communautaires, observé l'utilisation de nos ressources en classe et fait une itération constante du programme pour le mettre à l'épreuve.

✘ [enseignants en programmation](https://enseignantsenprogrammation.com)

NOTRE FORMULE

# pour bien enseigner la programmation



Notre contenu a été conçu spécialement pour les femmes dans le but de favoriser un environnement techno où règne l'égalité des genres.

De nombreuses études ont démontré que les filles voient les ordinateurs comme des outils servant principalement à accomplir des tâches, tandis que les garçons considèrent l'ordinateur comme un jouet ou une extension d'eux-mêmes. Chez Canada en programmation, nous n'enseignons pas seulement comment créer, mais pourquoi il est important de créer. Nous offrons l'occasion aux jeunes d'apprendre à propos de la technologie et de collaborer sur des projets de création.



**Notre contenu est à la fois amusant et stimulant.**

Nous souhaitons partager notre passion pour la technologie. C'est pourquoi, en collaboration avec des développeurs, des concepteurs et des éducateurs professionnels, nous avons créé du contenu qui explique des concepts de programmation de base et met en valeur des idées qui captiveront les apprenants.



**Les apprenants auront la possibilité d'apprendre et de collaborer avec des pairs qui partagent leurs champs d'intérêt.**

En réalisant des exercices pratiques en équipe, les apprenants sentiront qu'ils appartiennent à un groupe de jeunes débrouillards et curieux à propos de la technologie. Ainsi, leur apprentissage sera à la fois stimulant et gratifiant.



**Nous souhaitons épanouir les apprenants et les aider à avoir confiance en eux-mêmes.** Les cours et les exercices de d'Enseignants en programmation ont pour objectif d'orienter les jeunes vers la croissance. Nous leur enseignons comment apprendre de leurs erreurs et développons leur confiance en soi et leur autonomie en les incitant à apprendre et à créer.

# enseignants en programmation



**Le coffre à outils numérique de d'Enseignants en programmation comprend des guides, des leçons et des ressources conçus pour fournir aux éducateurs les connaissances générales, le savoir et la confiance nécessaires pour initier les apprenants à la programmation.**

Le Guide de démarrage que vous lisez en ce moment comprend des informations générales qui vous aideront à enseigner la programmation en classe. Il a été rédigé pour être utilisé conjointement avec les guides propres aux outils de programmation, comme Scratch, Mozilla Thimble et les lunettes à rayons x Mozilla. Ces guides propres aux outils comprennent des plans de cours détaillés et des activités de programmation prêtes à être intégrées à vos cours!

## CONSEILS POUR ENSEIGNER LA PROGRAMMATION TOUT EN S'AMUSANT

**1** Assurez-vous d'être à l'aise avec l'outil, mais ne vous inquiétez pas si vous ne le maîtrisez : laissez les apprenants s'aider entre eux.

**2** Ayez une idée claire de ce que vous souhaitez accomplir et trouvez un collègue qui aimerait enseigner à vos côtés. La clé est d'intégrer l'apprentissage de la programmation à votre enseignement.

**3** Adoptez une approche axée sur la croissance et encouragez les apprenants à apprendre de leurs erreurs. Rappelez au groupe que les compétences ne s'acquièrent pas seulement avec du talent, mais en travaillant dur et en faisant preuve de persévérance.

**4** Invitez des experts, des conférenciers et des bénévoles de votre région qui donneront de petites leçons et offriront un soutien supplémentaire à l'apprentissage de la programmation.

**5** Laissez les apprenants et leur créativité orienter votre cours. Que souhaitent-ils approfondir? Que souhaitent-ils découvrir?

**6** Faites appel à votre imagination, n'ayez pas peur de faire des erreurs et, surtout, amusez-vous!



ENSEIGNER LA PROGRAMMATION

# pour les débutants



## Qu'est-ce que le code?

L'explication la plus simple est la suivante : un ensemble de directives données à un ordinateur pour exécuter une tâche précise. L'ensemble de ces directives constitue un algorithme.

Les ordinateurs comprennent les directives très littéralement. Par conséquent, les directives ou algorithmes compris dans notre code doivent être très précis. Par exemple, en lisant une recette pour préparer un sandwich au beurre d'arachides et à la confiture (un algorithme), un humain interpréterait « mettre la confiture sur le pain » comme une directive, celle d'étaler de la confiture sur une tranche de pain. Toutefois, un robot interpréterait cette directive différemment : il mettrait le pot de confiture sur le dessus du sac de pain. Pour les humains, il s'agit d'une directive facile à comprendre, mais pour les ordinateurs, c'est une autre paire de manches.

Afin que l'ordinateur comprenne, il faudrait lui donner les directives suivantes :

1. Tourner le couvercle du pot de confiture dans le sens horaire jusqu'à ce que le couvercle soit complètement dévissé.
2. Soulever le couvercle du pot de confiture vers le haut.
3. Placer le couvercle du pot de confiture à côté du pot de confiture.

Et ainsi de suite.

Le code est extrêmement polyvalent : il sert à donner des directives aux robots, à créer des pages Web, à concevoir des jeux vidéo, à analyser des ensembles de données, et plus encore. Les possibilités sont infinies. Bien que les langages de programmation soient utilisés pour réaliser des tâches distinctes (p. ex. : le HTML et le CSS pour la création de sites Web, le R pour l'analyse de données, le JavaScript pour des robots volants), ils ne sont que des manières différentes d'entrer des directives ou un algorithme dans l'ordinateur. ✖

# Pourquoi enseigner la programmation

La technologie est partout et elle n'est pas prête de disparaître. Les sciences, la technologie, l'ingénierie et les mathématiques (STIM) sont des disciplines d'avenir, particulièrement lorsqu'elles sont jumelées à d'autres domaines. Nous voulons donner aux jeunes du Canada les compétences essentielles dont ils ont besoin pour vivre dans notre société actuelle et s'épanouir dans celle du futur.

## Mais pourquoi enseigner la programmation?

Selon le Conseil des technologies de l'information et des communications du Canada, il y aura une pénurie d'environ 200 000 travailleurs dans le domaine des TIC d'ici 2020. Pour nos jeunes, l'apprentissage de la programmation peut mener à une carrière gratifiante et payante. Toutefois, l'enseignement de la programmation aux enfants ne sert pas qu'à les aider à comprendre la technologie et à obtenir un emploi stable dans l'avenir. Fondamentalement, elle développe leurs compétences en résolution de problèmes et en pensée critique.

### **Nous croyons qu'il est essentiel que les jeunes apprennent à programmer pour les raisons suivantes :**

#### **La programmation est un superpouvoir.**

L'apprentissage du code permet aux enfants de ne pas seulement consommer, mais de créer la technologie qui les entoure, comme les jeux vidéo, les sites Web et les robots.

**La programmation pousse les enfants à penser de manière différente.** L'apprentissage de la programmation aide les enfants à développer des compétences essentielles applicables dans tous les domaines. Grâce à elle, ils peuvent résoudre des problèmes touchant à la programmation ou à d'autres domaines en innovant.

#### **La programmation aide les enfants à mieux comprendre le monde qui les entoure.**

Si nous enseignons la biologie et les mathématiques aux enfants afin de les aider à comprendre le monde, il va de soi que nous devrions aussi leur enseigner comment les ordinateurs communiquent et comment communiquer avec eux.

**La programmation peut changer le monde.** Si nous enseignons aux enfants comment utiliser la technologie comme outil de création, ils pourront créer des solutions à des problèmes de tous les jours.

**La programmation est amusante!** Nous voulons que les enfants ressentent la satisfaction d'avoir créé quelque chose eux-mêmes.

# Question de diversité

Il est indéniable que le secteur de la technologie manque de diversité. Nous avons donc conçu nos programmes et nos leçons en mettant l'accent sur la diversité. Pour régler la problématique du manque de diversité dans le secteur, il est important d'encourager des groupes d'apprenants hétérogènes et de répondre à leurs besoins.

## CONSEILS POUR MOTIVER TOUT LE MONDE

### **Faites des liens entre la programmation, la technologie et des projets de création intéressants pour les apprenants.**

Choisissez des sujets pertinents ou adaptez les exercices afin qu'ils reflètent les champs d'intérêt des apprenants et les caractéristiques de la communauté. Vos apprenants aiment-ils la musique? Les arts? Les animaux? La nourriture? La philanthropie? Y a-t-il un problème connu des apprenants pouvant être réglé avec une solution techno?

### **Faites des liens entre la programmation, la technologie et des professions qui intéressent les apprenants et sont axés sur la création.**

Il n'y a pas qu'une profession en technologie, mais des milliers! Par exemple, les concepteurs travaillent sur des jeux, des films, des appareils médicaux, et plus encore. En leur faisant découvrir ces différents emplois de création, vous contribuez à approfondir leurs connaissances à propos du secteur.

### **Favorisez un environnement d'apprentissage où règnent la collaboration et un esprit de communauté.**

Encouragez le travail en équipe, le mentorat par les pairs et mettez en valeur les réussites et les efforts des apprenants. Organisez des activités pour briser la glace et des jeux pour créer des liens entre les apprenants, surtout s'ils ne se connaissent pas déjà.

### **Recrutez des mentors reflétant la diversité du groupe.**

Il est important que les bénévoles ou les conférenciers invités du secteur soient représentatifs du groupe afin que les apprenants puissent s'identifier à eux. La diversité des mentors est essentielle, car elle met en valeur la contribution de personnes appartenant à un groupe sous-représenté dans le secteur des technologies.

### **Demandez-vous « pourquoi », pas seulement « comment ».**

Rappelez aux apprenants les raisons pour lesquelles la programmation est importante; ce n'est pas que du code. Faites des liens entre les notions apprises et la réalité de vos apprenants à l'extérieur de la classe. Les jeunes filles ont particulièrement besoin de savoir que leur travail a une influence positive et peut changer le monde.

### **Méfiez-vous des tendances inconscientes.**

Prenez conscience de vos propres tendances ou préjugés. Inconsciemment, nous orientons les garçons vers des « choses de garçon » et les filles, vers des « choses de fille ». Notre société est remplie de préjugés subtils qui ont une influence sur la façon de penser des apprenants. Par exemple, il y a une fausse croyance selon laquelle « les filles ne sont pas bonnes en maths ». N'oubliez pas que certains apprenants ont aussi des préjugés. Si vous en entendez, n'hésitez pas à les contredire.

### **Reconnaissez les signes du syndrome de l'imposteur.**

Le syndrome de l'imposteur est très présent chez les membres des groupes sous-représentés dans la technologie. Il est souvent associé aux apprenants très performants. Les jeunes atteints du syndrome de l'imposteur ont l'impression de ne pas se sentir « assez bons ou intelligents » ou de ne « pas mettre assez d'efforts pour réussir ». Si vous avez des jeunes qui démontrent des signes du syndrome, offrez-leur du soutien supplémentaire et renforcez leurs aptitudes et compétences dans des disciplines comme les maths, les sciences et la technologie. ✖

# Objectifs d'apprentissage

**Il faut voir l'apprentissage de la programmation comme étant avant tout une façon de résoudre un problème plutôt que comme l'apprentissage d'une langue ou d'un outil en particulier comme le Java ou Scratch. Les langages de programmation évoluent et changent constamment. Toutefois, l'essentiel de la méthode de résolution de problème à l'aide d'un ordinateur reste le même.**

La pensée informatique, le processus de résolution de problème, peut être enseignée sans même toucher à un langage de programmation en particulier. Les enseignants ne doivent pas oublier que le code n'est que l'outil facilitant la résolution d'un problème d'une certaine manière. Il n'est pas nécessaire de connaître la syntaxe d'un langage par cœur pour l'enseigner : il suffit de comprendre la logique de la résolution de problème. Vous avez déjà très probablement appris comment résoudre des problèmes dans le cadre de votre parcours professionnel. Le temps est venu de faire un lien entre cette logique et l'enseignement de la programmation.

## En enseignant la programmation, nous enseignons aussi :

### La pensée informatique

- Le raisonnement logique;
- La pensée critique;
- La reconnaissance des formes;
- La résolution de problèmes complexes et leur division en petites parties;
- Le débogage;
- Le développement des idées, du concept initial au résultat final.

### Les concepts de base de l'informatique

- Les programmes sont créés par des humains et contiennent des ensembles de directives disant exactement à l'ordinateur quoi faire.
- Les ordinateurs ne sont pas intelligents et n'ont aucune intuition. Ils ne comprennent pas les choses de la même façon que les humains. Ainsi, il est important de leur donner des directives précises et exactes afin qu'ils puissent bien suivre les étapes.

- Il n'est pas nécessaire d'être expert en informatique pour rédiger du code. Néanmoins, il faut savoir comment le rédiger clairement et prendre le temps de bien réfléchir aux directives.

### La citoyenneté numérique

- Encourager les apprenants à être enthousiastes à l'idée de créer avec la technologie, plutôt que de tout simplement la consommer.
- Pousser les apprenants à « regarder ce qui se cache sous le capot » et à poser des questions sur la technologie qu'ils consomment.

### Créativité et collaboration

- Réflexion sur la conception;
- Innovation.



## Concepts de programmation

Il existe des centaines de langages de programmation. Bien qu'ils ne se ressemblent pas du tout à l'œil nu, ils sont fondamentalement identiques. Les concepts de base et les façons d'interagir avec un ordinateur sont les mêmes. Nous utiliserons des outils comme Scratch et Mozilla Thimble pour enseigner ces concepts aux enfants de manière amusante et pertinente.

- Variables
- Données
- Événements
- Séquences
- Itération et boucles
- Expressions conditionnelles
- Fonctions
- Parallélisme
- Logique booléenne
- Nombres aléatoires
- Débogage
- Notions de maths intégrées



### L'HISTOIRE ET LA CULTURE DU CANADA



Nous avons créé des leçons qui abordent différents aspects de notre histoire. Les participants apprendront à propos d'événements historiques et découvriront des histoires de chez nous de manière tout à fait novatrice. Nous avons conçu ces leçons afin qu'elles complètent le programme de formation de l'école, ce qui permettra aux enseignants de satisfaire aux objectifs pédagogiques tout en intégrant la programmation. ✂

# sur la pensée informatique

La pensée informatique consiste à déterminer la nature d'un problème et à préparer une solution qui sera comprise par un ordinateur (ou un autre humain). Il s'agit de diviser de gros problèmes en petits problèmes et de penser aux étapes nécessaires pour surmonter ces petits problèmes. Les concepts de pensée informatique ne sont pas propres à la programmation. Vous remarquerez que ce sont des principes de résolution de problème auxquels nous faisons appel dans la vie de tous les jours.

## Voici les concepts principaux de la pensée informatique :

- ✓ **Le raisonnement logique**  
Les ordinateurs n'ont pas d'intuition. Ils sont prévisibles. Le raisonnement logique permet aux enfants de comprendre la réaction d'un ordinateur. Ainsi, les enfants savent qu'en cliquant le bouton « X » rouge de leur navigateur, la fenêtre du navigateur se fermera. Le raisonnement logique vient naturellement chez les enfants. Ils développent un modèle mental du fonctionnement de la technologie très tôt dans leur vie.
- ✓ **Les algorithmes**  
Les algorithmes sont un ensemble de directives disant à l'ordinateur de réaliser certaines tâches. Les algorithmes sont partout dans notre vie : un plan de cours est un algorithme pour une leçon tandis qu'une recette est un algorithme pour préparer notre plat préféré. En écrivant des directives par étape en français, avec des phrases, nous faisons ce que l'on appelle du « pseudo code ».
- ✓ **La décomposition**  
Le code est complexe, tout comme le sont les problèmes. Un élément essentiel de la pensée informatique et de la programmation est la division d'un problème en petites parties faciles à gérer, comme quand nous divisons un résumé de livre en sections différentes.
- ✓ **L'abstraction**  
Après avoir divisé des problèmes en petites parties, nous en déterminons les éléments importants grâce à l'abstraction. L'abstraction nous permet de gérer la complexité d'un problème, comme lorsque nous choisissons les informations essentielles pour la résolution d'un problème en maths ou en français.
- ✓ **Les formes et les généralisations**  
Les formes et les généralisations constituent une partie importante de l'apprentissage de la programmation. Elles nous permettent de créer des règles et de résoudre des problèmes généraux. Par exemple, en mathématiques, une formule est une généralisation servant à résoudre différents problèmes.

# de la programmation

Ce glossaire des concepts clés vous aidera à mieux comprendre la programmation. Vous remarquerez que plusieurs de ces concepts existent dans d'autres domaines et dans la vie quotidienne. Ils ne sont donc pas aussi intimidants que vous le pensiez!

**Algorithme** : ensemble d'étapes à suivre pour résoudre un problème.

**Boucle** : élément exécutant une séquence plus d'une fois (p. ex. : le bloc « répéter indéfiniment » dans Scratch).

**Débugage** : trouver des problèmes dans du code et les résoudre.

**État** : caractéristique d'un élément. Le sens en programmation est le même que dans les autres domaines (p. ex. : la télé est allumée ou éteinte). Certaines variables ont des états, d'autres n'en ont pas. Par exemple, 42 veut dire 42, et c'est tout.

**Événement** : action entraînant une autre action (p. ex. : le bloc « quand le drapeau vert est cliqué » dans Scratch).

**Expression conditionnelle** : élément prenant une décision selon des conditions (p. ex. : s'il pleut, alors il faut ouvrir son parapluie).

**Fonction** : procédure exécutant une action distincte. Il existe des fonctions préfabriquées, comme le bloc « Si l'élément touche le bord, rebondir », qui comprennent un ensemble de commandes à exécuter.

**Logique booléenne** : calcul algébrique faisant appel à des opérateurs comme « et », « ou »

ou « non ». Les valeurs comprises dans le calcul doivent être fausses ou vraies (p. ex. : Si j'ai chaud ET que je suis au sec, ALORS, je me sens bien.)

**Modularisation** : exploration des liens entre les parties et le tout.

**Opérateurs** : expressions mathématiques et logiques (p. ex. le bloc « X et X » dans Scratch).

**Parallélisme** : caractère d'actions se déroulant en même temps.

**Remixage** : création d'une nouvelle idée ou d'un nouveau projet en modifiant une idée ou un projet qui existe déjà.

**Séquence** : série de commandes essentielles afin de réaliser une tâche. Les ordinateurs lisent et exécutent des commandes dans un ordre précis, de haut en bas.

**Syntaxe** : grammaire et orthographe d'un langage de programmation. Dans les langages à blocs comme Scratch, il n'y a pas de syntaxe, car les blocs constituent la structure.

**Tableau** : variable spéciale enregistrant plus d'une valeur à la fois. Les éléments sont numérotés afin que nous puissions nous y référer plus tard. Par exemple, un tableau appelé « chiens » pourrait contenir les éléments suivants : caniche, chihuahua et labrador.

**Variable** : enregistre une information changeant au fil du temps (p. ex. : le pointage d'un joueur mis à jour dans un jeu).

# Planification des cours

Il y a de multiples façons d'enseigner la programmation! Vous pourriez commencer par un club parascolaire ou intégrer la programmation à vos cours déjà planifiés. Par exemple, plutôt que de demander aux apprenants d'écrire un compte-rendu d'un événement ou personnage historique, vous pourriez plutôt leur demander de créer un site Web sur le sujet. Les possibilités sont infinies.

## Exigences minimales

### OBLIGATOIRE :

- Éducateur**  
Cette personne supervisera les apprenants, animera les séances et les exercices de programmation.
- Mentors bénévoles**  
Les mentors répondront aux questions des apprenants pendant que l'éducateur donne le cours. Nous vous conseillons de proposer ce rôle à des apprenants avancés. Facultatif, mais très recommandé.
- Matériel**  
Ordinateurs portables ou de bureau (au moins un ordinateur par paire d'apprenants).
- Rallonges et multiprises**  
Projecteur, écran de projection, adaptateurs de connexion (Vous pouvez aussi présenter votre écran aux apprenants d'une autre manière.)
- Apprenants**  
Vous trouverez une affiche à imprimer dans la section des ressources qui vous aidera à faire la promotion de la programmation.
- Internet**  
Connexion Wi-Fi pour tous les ordinateurs portables et de bureau utilisés en classe. Facultatif. Certains outils (p. ex. : Scratch) peuvent être utilisés sans connexion Internet. Toutefois, il y a du contenu dont l'accès nécessite une connexion (p. ex : lunettes à rayons x Mozilla).
- Salle**  
Un lieu de rassemblement!



ET DU PLAISIR!

# Préparation horaire



MOMENT	TÂCHE
<b>1 semaine avant la séance</b>	Aller à l'adresse <a href="https://enseignantsenprogrammation.com">enseignantsenprogrammation.com</a> et : <ol style="list-style-type: none"><li>1. Lire le plan de leçon (de Description jusqu'à Prolongement).</li><li>2. Consulter l'exemple de projet.</li><li>3. S'exercer à faire les exercices avec le corrigé.</li><li>4. Réviser le contenu des vidéos et les informations supplémentaires afin de maîtriser le sujet ou le thème de la leçon (s'il y a lieu).</li></ol>
<b>3 jours avant la séance</b>	Créer un compte sur les sites utilisés (s'il y a lieu).
<b>Veille de la séance</b>	Imprimer une copie du plan de cours et du corrigé afin de les avoir sous la main en cas de problèmes avec le Wi-Fi.  S'assurer d'avoir tout le matériel.
<b>Le jour même de la séance</b>	Installer et mettre en place le matériel nécessaire pour la séance.  Écrire les informations de connexion sur une feuille ou un tableau afin que tout le monde puisse les lire facilement.

# Donner le ton

Avant de commencer, il est important de donner le ton au cours en définissant des concepts et des valeurs clés pour orienter l'apprentissage des apprenants. Nous voulons encourager chez eux un état d'esprit axé sur la croissance, c'est-à-dire une volonté d'aller au-delà de l'échec. Pour créer un climat favorable, il est essentiel de préciser dès le début que les problèmes avec la technologie font partie du processus d'apprentissage. Encouragez les apprenants à explorer, à essayer, à faire des erreurs et à résoudre des problèmes en collaborant. L'aspect le plus important est de faire preuve de patience. L'apprentissage de la programmation est comme celui d'une langue, car il demande beaucoup de temps, de patience et de répétition.

## Voici quelques sujets que nous abordons dans tous nos programmes :

### Pépins avec la technologie

Si un ordinateur ne fonctionne pas bien et qu'un apprenant ressent de la frustration, dites-lui qu'il s'agit d'une situation courante. Essayez de régler le problème ensemble. Demandez-lui ce qu'il ferait si ce problème survenait chez lui. Fermerait-il et relancerait-il le programme? Redémarrerait-il l'ordinateur? Vérifierait-il si l'ordinateur est bien branché? Si le problème n'est pas réglé, cherchez la solution sur Google! Il est important de montrer à l'apprenant comment trouver des solutions pour régler des problèmes dans le futur.

### Résolution de problème centrée sur les apprenants

Encouragez les apprenants à poser des questions à d'autres personnes avant de venir vous voir en utilisant la règle « Trois avant moi ». Les apprenants connaissent souvent la solution à un grand nombre des problèmes liés à la technologie. Voyez la recherche d'une solution comme l'occasion parfaite pour développer leurs aptitudes en tant que meneurs et dirigeants d'un groupe. Vous pouvez également établir une liste de choses à

vérifier pour résoudre des problèmes avec les apprenants. Finalement, que fait-on quand rien ne fonctionne? On demande à notre ami Google! La débrouillardise et la capacité à faire des recherches sur Internet sont des compétences extrêmement importantes que nous vous encourageons à développer.

### Apprendre en posant des questions

Nous avons créé de nombreux exercices pour initier les apprenants aux concepts de programmation de base. Toutefois, nous vous rappelons qu'ils ne sont que des points de départ. Il est essentiel de laisser les apprenants s'interroger et suivre leurs idées pour favoriser leur apprentissage.



# de cours suggérée



Chaque leçon adopte la même structure :

- Introduction
- Programmation en groupe
- Activité

L'**introduction** donne le ton de la leçon en informant les apprenants sur le thème. Il s'agit du moment où vous devriez aborder les conseils de la section « Donner le ton ».

Ensuite, durant la section **Programmation en groupe**, vous présenterez l'outil utilisé (p. ex. : Scratch ou Thimble). Donnez aux apprenants l'occasion d'explorer l'outil et d'essayer des fonctions de manière structurée en faisant appel aux exercices. De plus, vous leur expliquerez les attentes quant à la résolution de problèmes et à la débrouillardise avec le code. Pour ce faire, demandez aux apprenants de trouver comment faire certaines actions plutôt que de leur donner la réponse directement. N'oubliez pas de verbaliser le raisonnement de votre démarche du point A au point B et de rediriger constamment les questions vers d'autres apprenants ou vers la référence pour trouver les réponses.

Finalement, vous ferez l'**activité**, qui est aussi la section la plus longue de la séance. Vous remarquerez que le début de cette section ressemble beaucoup à la programmation en groupe, car les apprenants tenteront d'abord de comprendre l'activité. Toutefois, la majorité du temps sera consacrée à la recherche d'idées pour remixer ou personnaliser leur projet. Pendant que les apprenants remixent leur projet, vérifiez souvent leur progrès et invitez des volontaires à l'avant de la classe afin que ceux-ci puissent partager leurs découvertes et leurs solutions avec le groupe.

Durant la dernière partie de l'activité, si possible, les apprenants présenteront leurs créations. Nous vous recommandons de faire venir chaque apprenant à l'avant de la classe pour qu'ils puissent présenter leur projet. Cependant, si vous manquez de temps, vous avez aussi l'option de présenter les projets sous forme de « galerie d'art ». Les apprenants laisseront leur ordinateur ouvert à l'écran de leur projet, rangeront leur chaise et se lèveront pour faire tour de la classe et admirer les créations de tout le monde.

N'hésitez pas à prendre une pause si votre groupe en a besoin. ✂



# Rythme



**Il est essentiel de faire preuve de flexibilité lorsque l'on enseigne à l'aide de la technologie. Parfois, vous serez interrompus ou la technologie ne fonctionnera pas comme prévu. Les apprenants ont aussi des expériences et des niveaux de compétences différents.**

Pour bien enseigner la programmation, il est important de déterminer les besoins du groupe. Votre objectif est d'enseigner des notions qui correspondent au niveau de la majorité des apprenants. Vous devrez aussi stimuler tout le monde, tant les apprenants avancés qui ont besoin d'un défi supplémentaire que les apprenants éprouvant des difficultés à suivre le reste du groupe et ayant besoin de consignes adaptées.

Chaque leçon comprend un corrigé avec les étapes nécessaires pour créer un projet de base. De plus, vous y trouverez des exercices supplémentaires et des défis pour les apprenants avancés. Pour soutenir les apprenants avancés, vous pouvez soit leur proposer des exercices de la liste ou leur fournir une copie du corrigé, qui contient des notions supplémentaires.

La section « Prolongement » du plan de cours a aussi été conçue pour les apprenants avancés.

De plus, pour éviter de perdre des apprenants qui ne travaillent pas au même rythme que le reste du groupe, ne présumez rien. Vérifiez la progression du groupe en posant des questions comme : « Qu'est-ce que... (p. ex. : un navigateur)? », ou « Que veut dire... (p. ex. : « interactif »)? ». Nous vous recommandons de demander aux apprenants de vous rappeler des notions comme les raccourcis de clavier. De plus, n'hésitez pas à répéter des informations afin que tout le monde puisse les assimiler.

Vérifiez la compréhension du groupe en regardant le visage des apprenants ou en leur demandant de faire un pouce vers le haut ou vers le bas pour connaître leurs impressions. Posez le bon type de questions. Plutôt que de leur demander « Est-ce que tout le monde comprend? Êtes-vous prêts à passer à la prochaine étape? », dites « Si tu as besoin d'une minute de plus, lève ta main. » ou « Si ton écran ne ressemble pas au mien, lève la main. » Plus vos questions seront précises, plus leurs réponses le seront aussi.

Idéalement, les apprenants termineront chaque leçon en ayant créé leur propre projet et l'auront présenté au groupe. Toutefois, si certains n'ont pas fini, aucun problème. Au bout du compte, nous souhaitons que les apprenants aient eu une expérience positive avec la technologie, et bien sûr, beaucoup de plaisir!

# des imprévus



Parfois, rien ne va plus. Nous perdons notre accès à Internet, les ordinateurs plantent et nous perdons notre travail. Quand cela arrive, nous sentons que nous avons échoué, et ce n'est pas agréable, tant pour les enseignants que pour les apprenants.

**Ci-dessous, vous trouverez des conseils pour résoudre les problèmes de technologie et vous y préparer.**

## PLANIFICATION

En ayant un plan solide pour les pépins techniques, vous serez confiant quand le cours commencera.

### Voici nos suggestions :

- Préparez les ordinateurs avant l'activité.
- Écrivez les identifiants et les mots de passe au tableau.
- Utilisez un seul identifiant plutôt qu'un par apprenant.
- Chargez les ordinateurs portables avant le cours.
- Encouragez les apprenants à former des équipes et à travailler ensemble.

## ACTIVITÉS PAPIER OU HORS LIGNE

Pas besoin d'Internet ou d'ordinateur pour enseigner la programmation ou la pensée informatique! Il existe de nombreuses activités hors ligne idéales pour enseigner des concepts de programmation essentiels. Consultez les activités hors ligne du coffre à outils numérique de Teachers Learning Code pour en savoir plus.

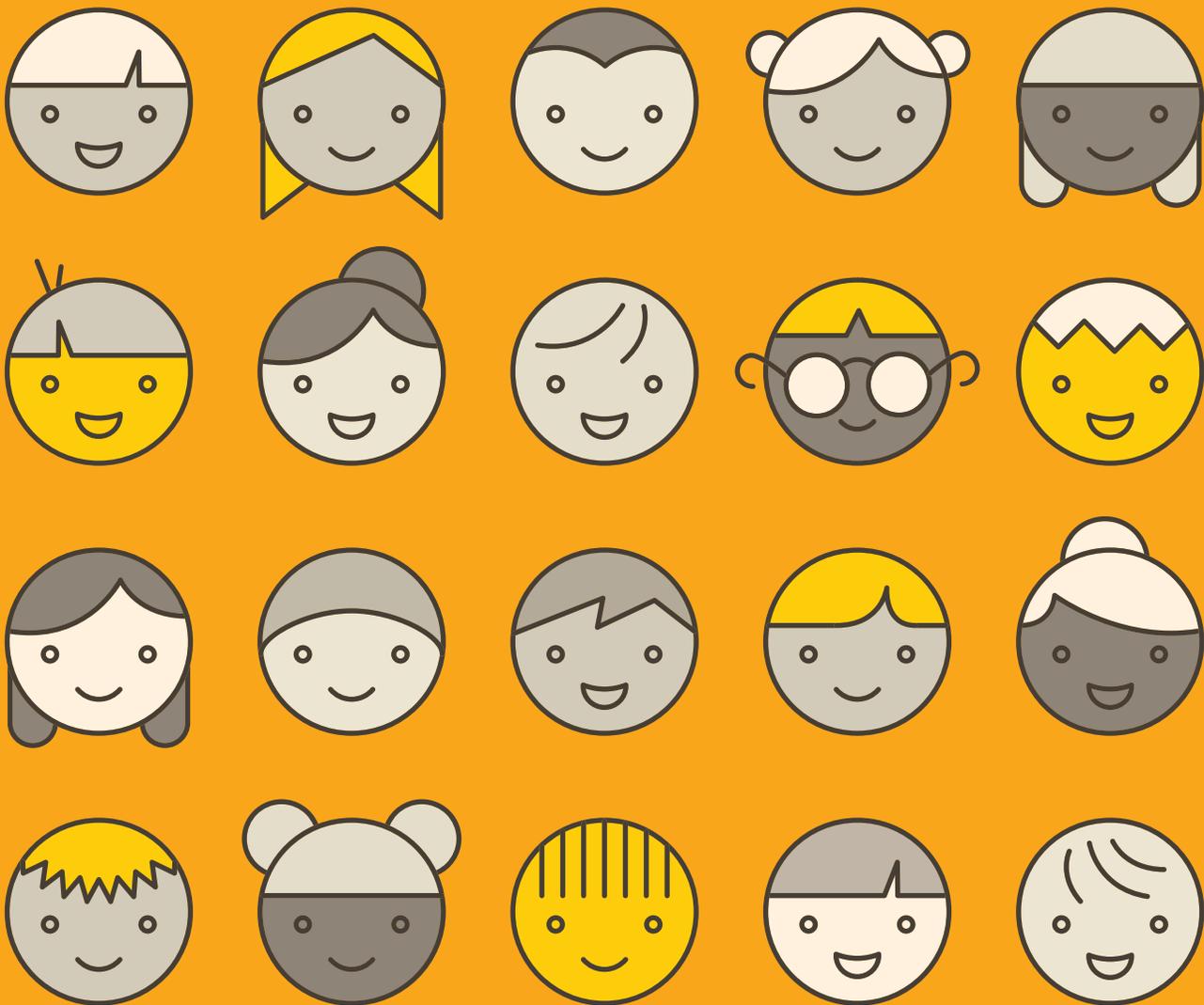












COMMUNIQUE AVEC NOUS

129 Spadina Ave., Suite 501  
Toronto, ON M5V 2L3

[info@canadalearningcode.ca](mailto:info@canadalearningcode.ca)  
[canadalearningcode.ca](http://canadalearningcode.ca)

✧enseignants en programmation