MODULE #1 | Découvrir la programmation créative KIT POUR LE DEUXIÈME TEMPS DE RENCONTRE

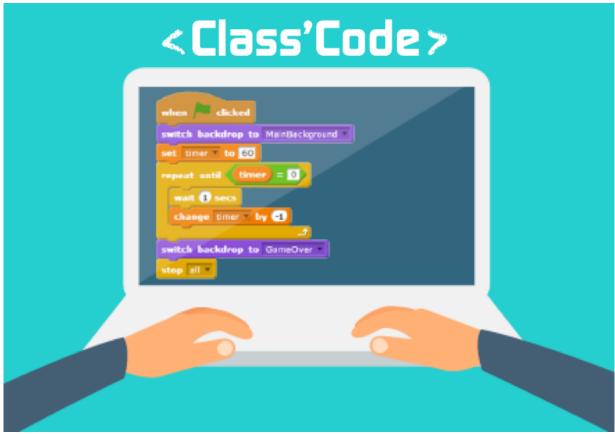


Image: OpenClassrooms, CC BY 4.0 International.

Le parcours de formation **Class'Code** s'articule autour de 5 modules de formation en ligne (MOOC) et des temps de rencontre (présentiel). Ce document a pour objectif de vous permettre de profiter au mieux de ce **deuxième** temps de rencontre autour du MOOC « **Découvrir la programmation créative** ».

Voilà maintenant trois semaines que vous avez fait vos premiers pas sur Scratch, mais aussi sur les algorithmes, instructions, boucles, variables, Grace Hopper, Abu Jafar, le robot, le jeu de Nim!

Comme pour la première séance, nous avons prévu une durée d'environ deux heures, mais la réalité sera sans doute variable d'un groupe à l'autre. Libre à vous de la suivre à la lettre, en désignant un maître du temps, ou de l'aménager en fonction de vos envies, de vos besoins et de vos contraintes. L'important étant de profiter de ce moment pour avancer.

N'hésitez pas, avant la rencontre, à utiliser le PAD mis à disposition sur la page du groupe pour faire remonter vos besoins, choisir les activités qui vous intéressent le plus et vous organiser. Cela vous permettra de profiter au maximum du temps de rencontre sans perdre trop de temps en début de séance.



QUE PRÉPARER AVANT LA SÉANCE?

Avoir suivi les trois premières semaines du MOOC sur openclassrooms.fr et être inscrit à un temps de rencontre sur classcode.fr.

Organisation

- Les activités sont prévues pour des groupes de 3 à 20 personnes.
- Faites remonter vos questions, besoins et activités préférées sur le PAD avant la séance.
- Prévoyez de créer des sous-groupes pour faciliter certaines tâches, notamment lors de l'entraide ou des activités débranchées.
- Les temps d'activités sont donnés à titre indicatif, si vous souhaitez tout réaliser durant la séance, nous vous conseillons de désigner un maître du temps qui veillera au respect des horaires :-)

Matériel nécessaire

- Ce kit pédagogique sous forme papier ou électronique (imprimez au moins la page 12 Annexe 2).
- Du **papier**, quelques **stylos** ainsi que **9 jetons** (pièces de monnaie, cailloux, etc.).
- Si possible, votre ordinateur portable ou tablette (idéalement il en faut un pour deux).
- Si vous n'être pas sûr d'avoir accès à Internet dans la salle qui accueille ce temps ce rencontre, pensez à sauvegarder vos **projets Scratch** sur votre machine (si nécessaire, revoir Partie 1, Chapitre 3).

CONCRÈTEMENT, QUE VA-T-ON FAIRE?

Pour profiter au maximum de ce temps de rencontre et de votre diversité, nous vous proposons des pistes d'activités qui s'articulent autour de quatre temps pour : se présenter, s'entraider, approfondir et échanger.

Se présenter - env. 15'

Accueil des nouveaux venus et retours sur les semaines écoulées.

S'entraider au codage et programmer ensemble - env. 30'

lci vous pourrez consacrer un moment à debugger vos programmes ou à réaliser un petit projet collaboratif, comme une sorte de mini code jam!

Approfondir - env. 35'-40'

Vous pratiquerez, ensemble, une activité débranchée (mise en situation). Cela vous permettra d'estimer la mise en pratique de l'activité (des adaptations sont-elles nécessaires, par exemple ?). Dans ce même temps vous reverrez certaines notions informatiques sous un angle différent.

Les concepts qui seront travaillés lors de ce temps de rencontre : les variables et des conditions. On va même plus loin, vous concevrez une activité pour apprendre à programmer une machine à café !

Échanger – au moins 30'

Et après, il va falloir transmettre. Discutez des enjeux en lien avec ce que vous venez d'apprendre, cherchez la meilleure façon d'expliquer ces notions à votre public, répondez collectivement à des questions, partagez vos idées.

Et... Vous pouvez lire la suite que lors de la séance, ou avant si vous préférez vous préparer, mais dans ce cas, sautez les rubriques « Éléments de réponses »! :)

MODULE #1 | Découvrir la programmation créative FEUILLE DE ROUTE

SE PRÉSENTER - env. 15'

Après quelques semaines de travail derrière l'écran, c'est le moment de nous retrouver. Nous nous connaissons déjà tous, ou presque... Si une nouvelle personne se joint à nous, prenons un moment pour l'accueillir. Nous pouvons faire un rapide tour de table de nos impressions depuis le dernier temps de rencontre. Sinon passons directement à la suite!

S'ENTRAIDER AU CODAGE ET PROGRAMMER ENSEMBLE - env. 30'

Coder à plusieurs permet de proposer un travail en équipe, de débrider notre imagination, mais aussi de voir comment chacun et chacune peut avoir des stratégies différentes pour résoudre un même problème, surtout quand il s'agit de codage informatique!

COMMENCER À CRÉER EN ÉQUIPE UN PROJET COLLABORATIF EN SCRATCH

Matériel et organisation

- Du papier et un stylo, pour réfléchir.
- Un ordinateur ou une tablette pour programmer.
- Se répartir en groupe de 2 à 4 et choisir ensemble : faire un remix ou un nouveau projet (voir les règles).
- Prévoir un temps pour se montrer les projets réalisés ou les mettre en ligne sur Scratch Studio pour les visionner plus tard.

Labyrinthe, le remix

Pourquoi pas remixer ensemble nos labyrinthes ? Nous pouvons partir de nos propres labyrinthes avec plusieurs niveaux ou encore de celui proposé en ligne (https://scratch.mit.edu/projects/100028393). Nous pouvons ajouter encore un niveau, de nouvelles conditions qui font gagner ou perdre des points, des vies, une animation qui félicite ceux qui arrivent au but... à nous de jouer!

Puis chaque sous-groupe, prend deux à trois minutes (pas plus) pour montrer le projet « en action ».

Pour aller plus loin

Si l'un d'entre nous a un projet à coder bien défini (vive la créativité!) et d'autres sont prêts à participer, pourquoi ne pas profiter de cette rencontre pour le démarrer, le partager sur Scratch Studio et le poursuivre même après la fin du Module?

RAPPEL POUR DÉBUGGER

Nouveaux codes, nouveaux bugs. Sans y passer toute la séance - ce serait dommage de ne pas profiter de ce temps de rencontre pour expérimenter et échanger avec les autres sans écran - vous aurez peut-être besoin d'un nouveau regard pour dépasser un blocage. Retrouvez ici les principaux conseils.

- Utiliser le forum de Scratch : scratch.mit.edu/discuss/15/
- Consulter la page « Comment ne pas se faire piéger pendant une animation Scratch? » : pixees.fr/?p=4252
- Regarder la question de la FAQ facilitateur « Comment aider quelqu'un à trouver une erreur de programmation ? » : pixees.fr/?page_id=8012
- Toujours bloqué ? En dernier recours contacter le bureau d'accueil de Class'Code : pixees.fr/?page_id=42

APPROFONDIR - env. 35'-40'

Nous allons approfondir la compréhension de certains concepts informatiques que nous avons travaillés ces trois premières semaines. Nous allons revoir les mêmes notions, en variant les activités possibles.

ACTIVITÉ 1 | MAIS ÇA CHANGE TOUT LE TEMPS ? (les variables)

Matériel

- Quelques jetons (pièces de monnaie, cailloux -> au moins 9) et quelques stylos.
- L'annexe 2 (page 12) de ce kit. Voici notre espace de stockage!

Règles

- Disposons l'annexe 2 (page 12) sur une surface de façon à ce que tous puissent les voir.
- Attribuons une lettre à chaque rectangle (A, B, C, etc.). Voilà la mémoire de la machine, son espace de stockage.
- Tout comme une machine, nous allons suivre des instructions. Nous allons aussi, à chaque fois, « traduire » ces instructions à haute voix, par exemple :

$A \leftarrow 2$ Remplacer le nombre de jetons actuellement sur le rectangle A par 2.

• On remarque le codage : la flèche vers la gauche attribue le nombre de jetons au rectangle correspondant, ceci est une convention. Voici un autre exemple :

• Vérifier le nombre de jetons actuellement sur le rectangle F.

• Vérifier le nombre de jetons actuellement sur le rectangle A.

- Soustraire mentalement le nombre de jetons actuellement sur le rectangle A de ceux qui sont actuellement sur le rectangle F (on ne change pas ce qui est sur A ou F).
- Remplacer le nombre de jetons actuellement sur le rectangle G par le résultat de cette soustraction.

 $G \leftarrow F - A$

 On y va? Tout comme une machine, suivons les instructions ci-dessous afin de distribuer les jetons sur les rectangles. Imaginons, par exemple, que le rectangle A sert à garder le nombre de vies d'un joueur, le rectangle 2 le niveau, etc.:

#1 | Comment traduire les instructions suivantes ?

A ← 2 B ← 1 C ← A − B D ← A + C E ← B + 1

#2 | Décryptage - adulte ou jeunes de 8 à 12 ans, répondons aux questions suivantes :

- Qu'est-ce qu'une variable ? Ici quel objet a la même fonction qu'une variable informatique ?
- Comment savoir **où trouver** l'information requise pour réaliser l'instruction 3 ? Quel élément permet de le faire ? Quel est le nom de ce processus en informatique ?
- La valeur attribuée à chaque rectangle ou « espace de mémoire » est bien sûr, le nombre de jetons sur le rectangle correspondant. Cette valeur, **est-elle fixe** ?

#3 | Que se passerait-il si on ajoutait la ligne ci-dessous?

A ← 3 + B

/// ÉLÉMENTS DE RÉPONSES

SOLUTION #1 | Comment traduire les instructions suivantes?

A ← 2	Remplacer le nombre de jetons actuellement sur le rectangle A par 2.	
B ← 1	Remplacer le nombre de jetons actuellement sur le rectangle B par 1.	
C ← A − B	 Vérifier le nombre de jetons actuellement sur le rectangle A. Vérifier le nombre de jetons actuellement sur le rectangle B. Soustraire mentalement le nombre de jetons actuellement sur le rectangle B de ceux qui sont actuellement sur le rectangle A (on ne change pas ce qui est sur A ou B). Remplacer le nombre de jetons actuellement sur le rectangle C par le résultat de cette soustraction. 	
D ← A + C	 Vérifier le nombre de jetons actuellement sur le rectangle A. Vérifier le nombre de jetons actuellement sur le rectangle C. Additionner mentalement le nombre de jetons actuellement sur le rectangle A à ceux qui sont actuellement sur le rectangle C (on ne change pas ce qui est sur A ou C). Remplacer le nombre de jetons actuellement sur le rectangle D par le résultat de cette addition. 	

Vérifier le nombre de jetons actuellement sur le rectangle B.

• Additionner mentalement 1 au nombre de jetons actuellement sur le rectangle B (on ne change pas ce qui est sur B).

 Remplacer le nombre de jetons actuellement sur le rectangle E par le résultat de cette addition.

SOLUTION #2 | Décryptage - adulte ou jeunes de 8 à 12 ans, répondons aux questions suivantes :

- Qu'est-ce qu'une variable ? Ici, quel objet a la même fonction qu'une variable informatique ?
 Un endroit (espace dans la mémoire de la machine) où on garde une valeur, une donnée. Cela correspond aux rectangles dans notre exemple.
- Comment savoir **où trouver** l'information requise pour réaliser l'instruction 3 ? Quel élément permet de le faire ? Quel est le nom de ce processus en informatique ?

 Nous savons où trouver le nombre de pièces dans chaque « espace de mémoire » grâce à l'**identification** que nous avons faite au préalable, en utilisant les lettres de l'alphabet (A, B, C, etc.). On appelle cela le processus de **déclaration**. C'est ce qu'on fait quand on crée une variable vie qui stockera le nombre de vies restantes d'un joueur.
- La valeur attribuée à chaque rectangle ou « espace de mémoire » est bien sûr, le nombre de jetons sur le rectangle correspondant. Cette valeur, est-elle fixe?
 La valeur stockée sur chaque rectangle correspond au nombre de pièces sur ce rectangle, à un moment donné. Remarquez que cette valeur est variable, selon les instructions qui sont fournies (c'est le concept d'affectation).

SOLUTION #3 | Que se passerait-il si on ajoutait la ligne ci-dessous?

 $E \leftarrow B + 1$

 $A \leftarrow 2$ $B \leftarrow 1$ $C \leftarrow A - B$ $D \leftarrow A + C$ $E \leftarrow B + 1$ $A \leftarrow 3 + B$

Quand on ajoute la dernière ligne au code, la valeur de A change. Attention, cela ne veut pas dire que les autres valeurs qui en dépendent (C et D) changent également. Il ne faut pas oublier :

- l'ordre des instructions fournies à une machine est important.
- les machines ne font que suivre des ordres. Si je ne lui dis pas (sous forme d'instruction) d'actualiser les valeurs de C et D par rapport à la nouvelle valeur de A, elle ne le fera pas toute seule.
- au début, quand vous avez commencé, les rectangles étaient vides!

Remarques: Nous avons remarqué que la valeur d'une variable peut être un nombre (comme dans cette activité), mais également un texte (le texte que le joueur fournit quand le lutin demande son nom sur Scratch). Remarquons également que les variables peuvent être affectées à travers le résultat d'un calcul (comme ici), mais aussi à travers une entrée de la part de l'utilisateur (comme quand Bobby nous demande notre prénom, ou que nous informons du nombre d'allumettes à prendre pendant le jeu de Nim).

ACTIVITÉ 2 | LA MACHINE À CAFÉ (créer une activité)

L'idée de cette activité est de réaliser le <u>chemin inverse</u> : nous vous proposons un algorithme et l'explication de son exécution par une machine ; <u>à vous ensuite d'imaginer une activité liée</u> et peut-être la proposer ensuite sur pixees.fr!

Objectif

<u>Concevoir et développer une activité débranchée</u> où un humain vient chercher du café et la personne qui mime la machine exécute l'algorithme qui lui rend la monnaie (et le café!).

Règles

Après avoir pris connaissance de la situation (algorithme) ci-dessous, nous allons prendre 10 minutes, par sous-groupes de trois ou guatre personnes pour :

- Choisir une tranche d'âge (minimale? pour des plus grands?).
- Décider de la durée de l'activité.
- Expliciter les préreguis dont les jeunes ont besoin.
- Construire un scénario le plus simple possible.
- Créer des jalons pour permettre aux jeunes d'avancer pas à pas vers la solution, ne pas oublier, il faut que ça soit fun! Pourquoi ne pas remplacer la machine à café par un distributeur de bonbons?

Situation de départ

Que se passe-t-il quand on achète un café... à une machine à café ? On entre des pièces pour payer le café et la machine nous fait un café! Si la somme fournie est plus importante que le prix du café, la machine nous rend la monnaie. Mais comment fait-elle pour savoir combien nous rendre ? Regardons les étapes qui se cachent dans cette situation du quotidien.

1. Entrer des pièces dans la machine

Nous avons besoin d'une variable pour stocker la valeur totale rentrée par l'usager. Nous l'appellerons ici valeur_rentree. Disons que cette variable a été prévue (et créée) par le fabriquant de la machine. Tout commence quand l'usager rentre une pièce de monnaie dans la machine :

QUAND [pièce rentrée] ALORS

valeur_rentree

valeur_rentree + valeur de la pièce rentrée

- QUAND est un événement. Nous avons déjà utilisé ce type d'instruction sur Scratch.
- On incrémente la somme déjà rentrée avec ce qui rentre dans la machine.

2. Distribuer le café

D'abord, il faut que le prix (actuel) du café soit disponible. Pour cela nous allons donc également utiliser une variable, identifiée prix par le fabriquant de la machine.

 $prix \leftarrow 0,50$

Quand l'usager appuie sur le bouton « Café », la machine va utiliser les valeurs conservées par les variables valeur_rentree et prix pour calculer la monnaie à rendre. Il faudra alors une troisième variable pour stocker le résultat de ce calcul, appelons-la difference :

difference ← valeur_rentree – prix

Ainsi, il y a 3 situations possibles :

SI [difference < 0] ALORS
afficher le montant manquant
SI [difference = 0] ALORS
faire le café
SI [difference > 0] ALORS
faire le café
rendre la monnaie (voir ci-dessous)

3. Rendre la monnaie

La possibilité la plus simple est de vérifier la valeur stockée dans la variable difference et d'énumérer tous les cas possibles, par exemple :

SI [difference = 0,10] ALORS
rendre 0,10
SI [difference = 0,20] ALORS
rendre 0,20
SI [difference = 0,30] ALORS
rendre 0,10
rendre 0,20
SI [difference = 0,40] ALORS
rendre 0,20
rendre 0,20
SI [difference = 0,50] ALORS
rendre 0,50

Sans oublier de (re)mettre à zéro les variables valeur_rentree et difference :

 $valeur_rentree \leftarrow 0$ $difference \leftarrow 0$

Remarque: Dans ce scénario, la situation a été simplifiée à des fins pédagogiques: il y a toujours assez de monnaie dans la machine et la solution est simple même si ce n'est pas la plus efficace. Pour une solution plus élégante, il aurait fallu « abstraire » davantage les données.

ÉCHANGER - au moins 30'

Et après, il va falloir transmettre. Discutons des enjeux liés à ce que nous venons d'apprendre, cherchons la meilleure façon d'expliquer ces notions à notre public, répondons collectivement à des questions posées en ligne, partageons nos idées!

Autour des concepts informatiques et du vocabulaire

Au cours de ces dernières semaines et de ce temps de rencontre, nous avons abordé de nombreuses notions. Un certain nombre de termes sont repris en annexe. Comment les définirions-nous ?

Dans le module en ligne (Partie 3, Chapitre 4) on vous propose de trouver autour de vous des machines qui peuvent manipuler les quatre ingrédients des algorithmes et donc d'exécuter n'importe quel algorithme. Quelles sont celles que vous avez identifiées ?

Autour de la transmission

La dernière activité en ligne (Partie 3, Chapitre 5) propose de commencer à préparer un atelier de découverte de la programmation créative. Pourquoi ne pas la partager ici avant de se lancer comme défi de la mettre en œuvre avec des jeunes de votre entourage ?

ET ENSUITE...

Vous avez presque fini le Module #1 de Class'Code, Découvrir la programmation créative! À l'issue des 4 semaines de formation en ligne et de ces deux temps de rencontre nous faisons le pari que vous serez capables d'animer un premier atelier d'initiation avec des jeunes.

C'est maintenant que tout commence! Vous avez sûrement encore des questions, des doutes, vous allez découvrir de nouvelles situations. Vous pouvez rester en contact et continuer à vous entraider et à échanger sur ces sujets! Vous faites maintenant partie de la communauté Class'Code.

N'hésitez pas à nous faire part de vos retours et avis : points positifs ou à améliorer, problèmes rencontrés, idées nouvelles, c'est avec vous que nous construisons cette formation.

Envie de poursuivre ? D'autres modules seront bientôt disponibles pour aller plus loin. Ils vous permettront d'animer des ateliers thématiques autour de l'information, de la robotique ludique et des réseaux. Retrouvez l'ensemble du parcours sur classcode.fr

L'équipe de Class'Code classcode-accueil@inria.fr



Ce kit < Class'Code > Module#1 : Découvrir la programmation créative | Kit pour le premier temps de rencontre de l'équipe Class'Code est mis à disposition selon les termes de la licence Creative Commons Attribution 4.0 International.

MODULE #1 | Découvrir la programmation créative ANNEXE 1 – VOCABULAIRE

Variable

C'est l'objet permettant d'associer une valeur à un identifiant. La variable peut être lue (par exemple, consulter la valeur que contient la variable ayant pour identifiant *vie* d'un joueur). Elle peut aussi être modifiée grâce à l'instruction d'affectation (par exemple, mettre la variable *vie* à 0 en début de partie, ou la variable *prenom* qui prend la valeur « Aïcha » ou « David » selon ce que le joueur tape quand on lui demande son prénom).

Identifiant

C'est le nom de la variable, c'est-à-dire *compteur* ou *prénom* dans les exemples précédents. On parle aussi d'étiquette. Cela permet de désigner la variable, donc indirectement l'espace de mémoire de la machine qui est utilisé.

Déclaration

C'est le processus qui permet de créer une variable, en réservant un espace défini dans la mémoire de la machine.

Valeur attribuée

C'est la valeur (à un moment donné) attribuée à la variable : 0,50 dans le cas du prix du café ou « Aïcha » dans le cas du prénom d'un joueur.

Affectation

C'est l'instruction qui détermine (ou change) la valeur attribuée à une variable.

Plus de définitions sur pixees.fr

MODULE #1 | Découvrir la programmation créative ANNEXE 2 – ESPACE DE STOCKAGE

	-	
	ĺ	
1	ĺ	
1	ĺ	
1	ĺ	
1	ĺ	
	ĺ	
1	ĺ	
	ĺ	
1	ĺ	
	ĺ	
1	ĺ	
1	ĺ	
	1	
1	ĺ	
	1	
	1	
1	ĺ	
	1	
	ĺ	
1	ĺ	
	ĺ	
1	ĺ	
	ĺ	
1	ĺ	
1	ĺ	
1	ĺ	
1	ĺ	
1	ĺ	
	1	
	Í	
1	ĺ	
1	ĺ	
1	ĺ	
	ĺ	
1	ĺ	
	1	
1	ĺ	
	1	
	ĺ	
	1	
	1	
	ĺ	
	1	