



Bonjour !

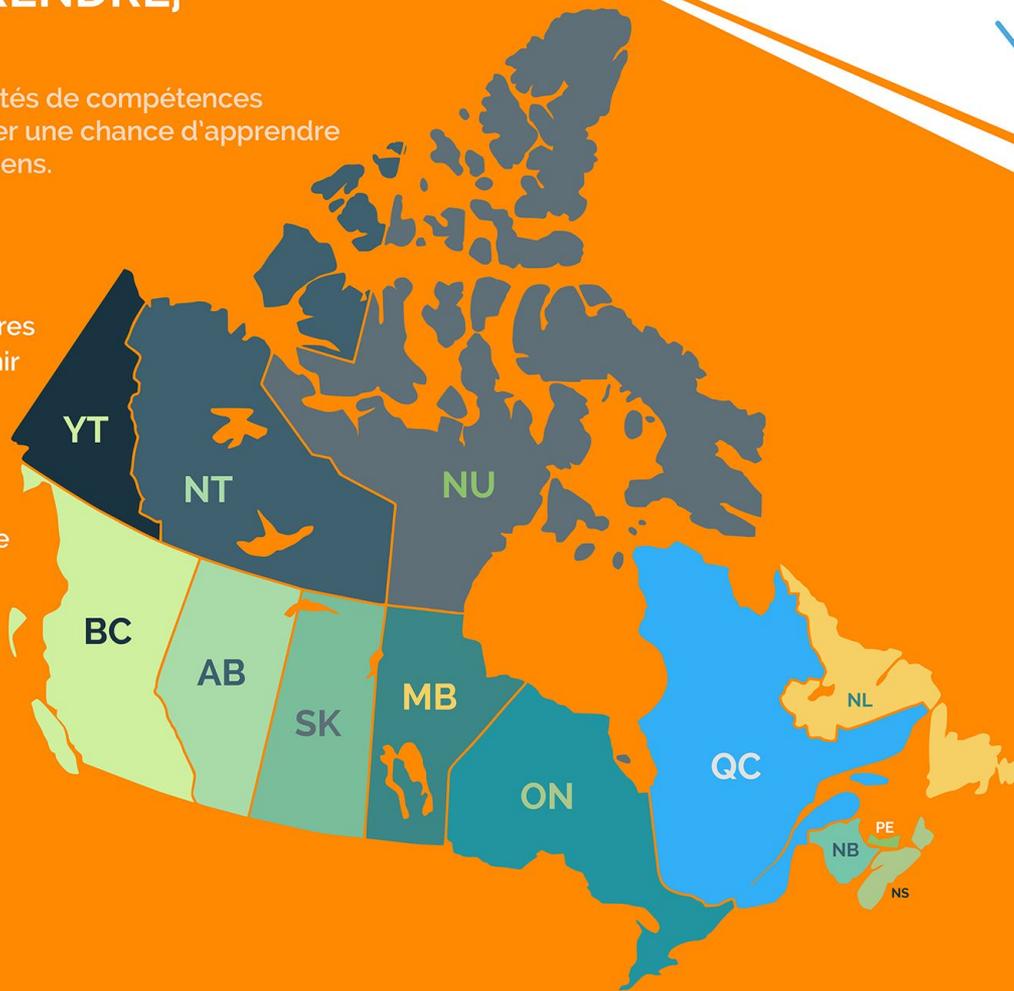
WIFI:

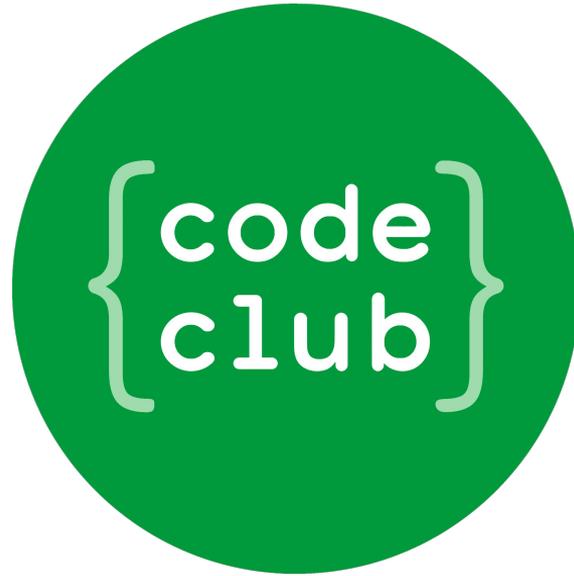
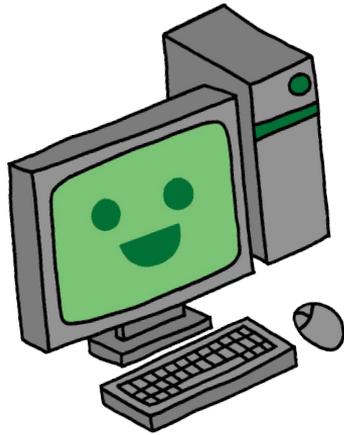


CODER, CRÉER, APPRENDRE, ENTHOUSIASMER

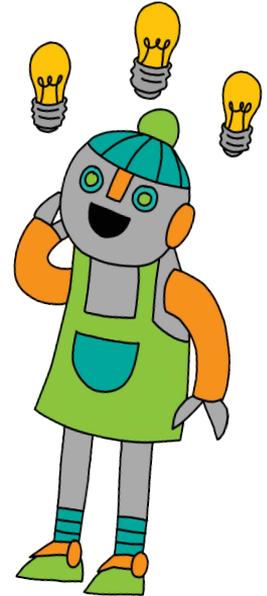
Créer et soutenir des communautés de compétences numériques durables pour donner une chance d'apprendre à coder à tous les enfants canadiens.

Le projet CodeCode de KCJ est une initiative regroupant plusieurs partenaires dont l'objectif est de créer et de soutenir des communautés de compétences numériques pour tous les enfants canadiens, qu'ils viennent de villes métropolitaines ou de régions rurales. Nous croyons que l'impact à long terme le plus durable est d'intégrer le code dans l'éducation, l'art, le sport, la science, le jeu ou la santé et dans de nombreux aspects de l'environnement de l'enfant. L'objectif est de les aider à devenir de meilleurs penseurs et de meilleurs créateurs.



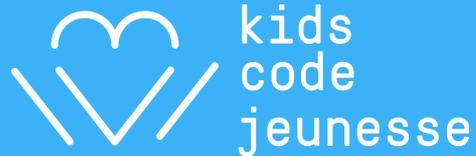
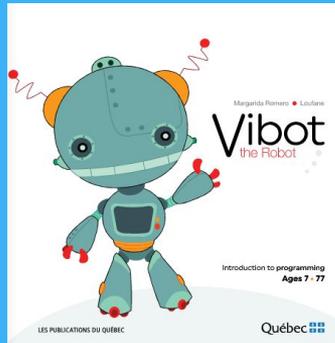


CANADA



PARTAGEZ VOTRE SUCCÈS AVEC NOUS!

Partagez une photo de l'atelier dès aujourd'hui et courez la chance de **gagner Vibot le robot**



@kidscoding



Consortium
provincial francophone

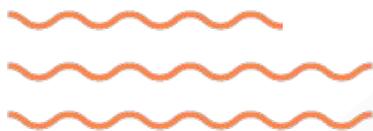


@cpfpp

#KCJalberta
#arpc
#CodeCan
#abed



Sauvetage Robot Ping Pong



Horaire

9h30 - Mot de bienvenue et Introduction

De 9 h 45 à 10 h 15 - Activité brise-glace / Activité débranchée

10h20 - Introduction à la pensée informatique

10h50 - Pause café

11h00 - Introduction à la pensée informatique

12h00 - Dîner

12h30 - Atelier Micro:bits

2h00 - Réflexions et Commentaires - C'est l'heure du sondage !



Pensée Informatique

Vous le faites déjà.

Qu'est-ce que la pensée informatique?

C'est une façon d'aborder la résolution de problèmes qui met l'accent sur la reconnaissance des schémas, la décomposition, l'abstraction et les algorithmes. Ce sont les concepts clés qui forment la logique derrière la pensée informatique. D'ailleurs, ce sont des pratiques quotidiennes que nous faisons tous sans les avoir étiquetées ainsi.

Décomposition

Réduire un problème **complexe** en morceaux **simples**

[Le petit chien] [a trouvé une famille d'adoption] [depuis une semaine].

↓
Sujet
(GN)

↓
(GN)

↙ ↓ ↘
Dét. GAdj Nom

↓
Prédictat
(GV)

↓
(GV)

↙ ↘
Verbe GN

↓
Complément de P
(GPrép)

↓
(GPrép)

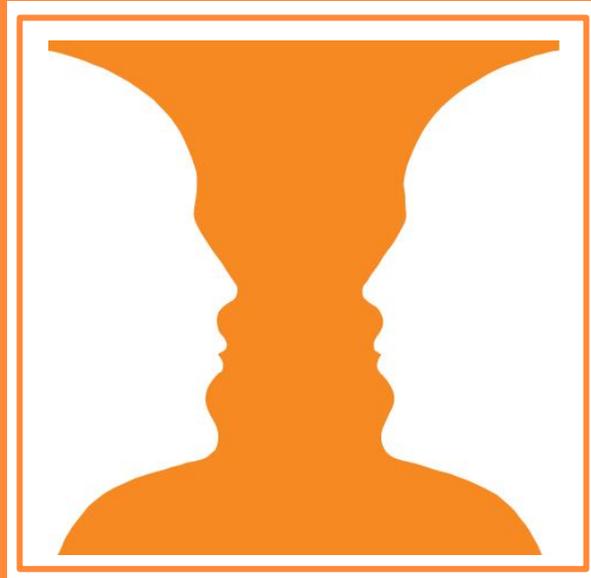
↙ ↘
GN Prép

Reconnaissance des schémas

Apprendre à identifier et se servir des similitudes pour: **simplifier, abrégé et appliquer des solutions semblables**

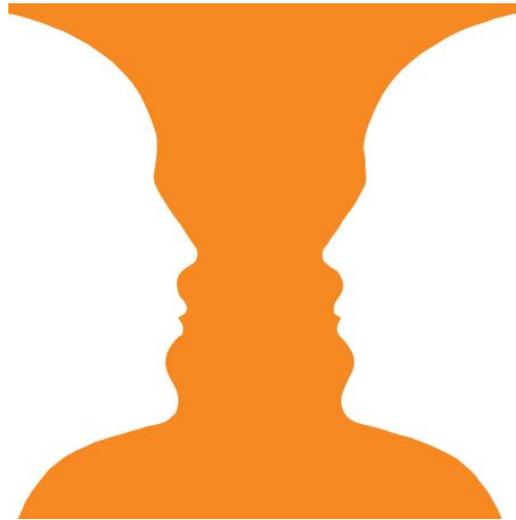
Reconnaissance des schémas

Apprendre à identifier et se servir des similitudes pour: **simplifier, abréger et appliquer des solutions semblables**



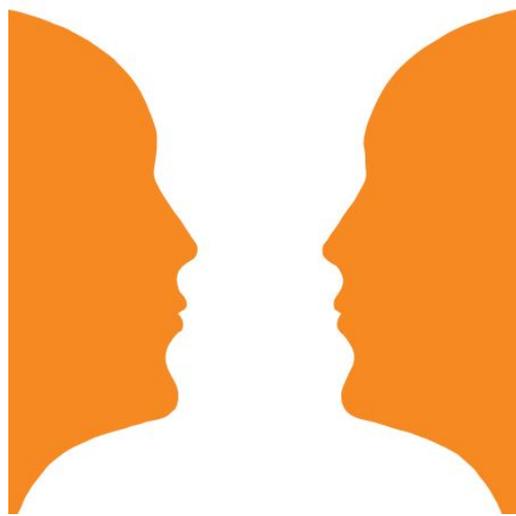
Reconnaissance des schémas

Apprendre à identifier et se servir des similitudes pour: **simplifier, abrégé et appliquer des solutions semblables**

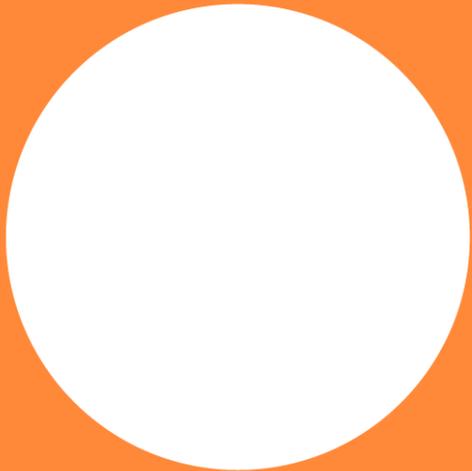


Reconnaissance des schémas

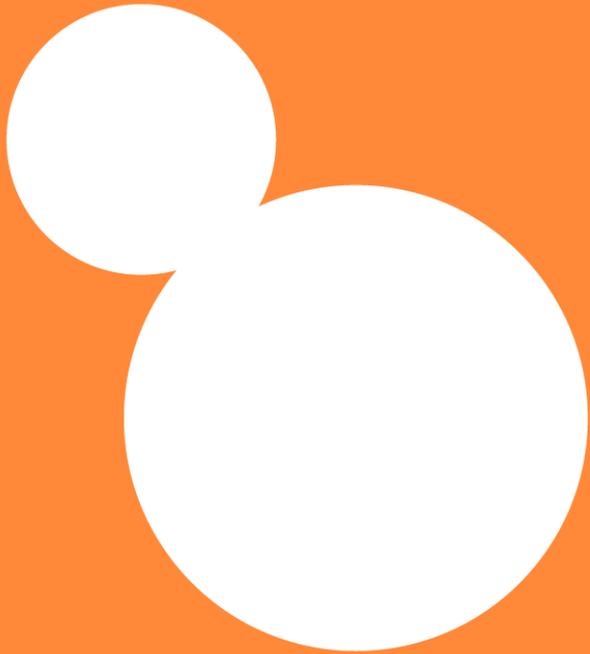
Apprendre à identifier et se servir des similitudes pour: **simplifier, abréger et appliquer des solutions semblables**



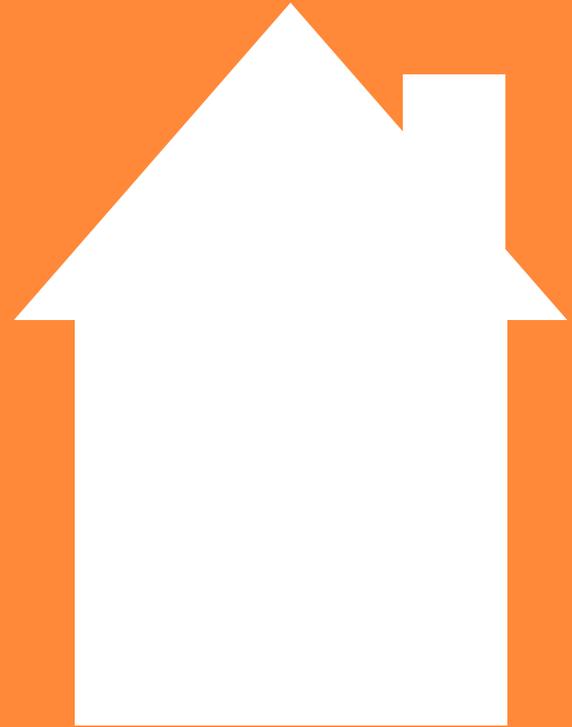
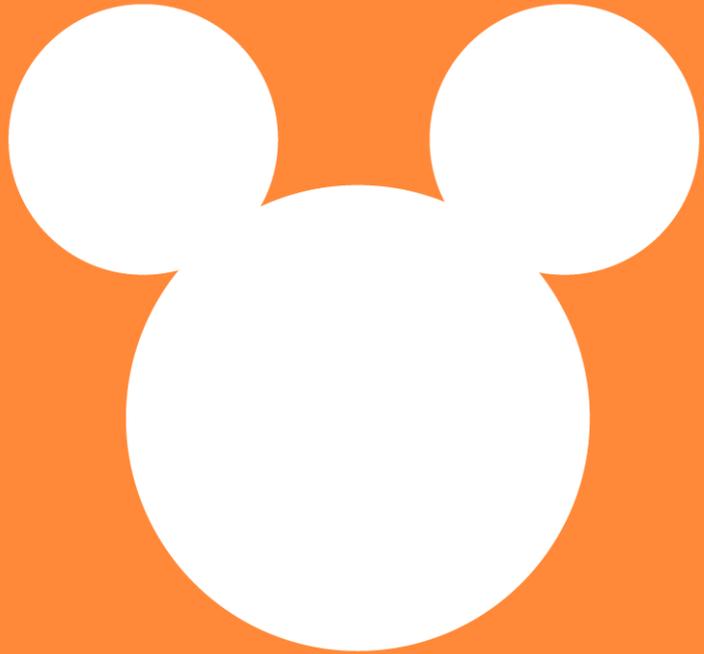
Reconnaissance des schémas



Reconnaissance des schémas



Reconnaissance des schémas



Abstraction

Réduire à un ensemble de caractéristiques essentielles

Résumer

Modèles non réalistes

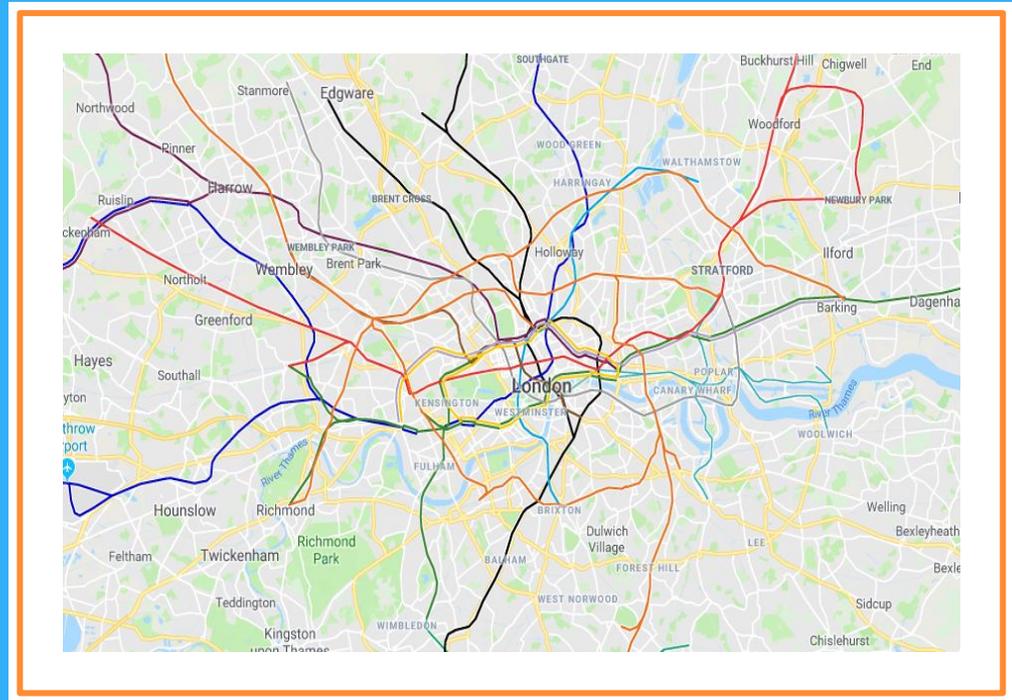


Abstraction

Réduire à un ensemble de caractéristiques essentielles

Résumer

Modèles non réalistes

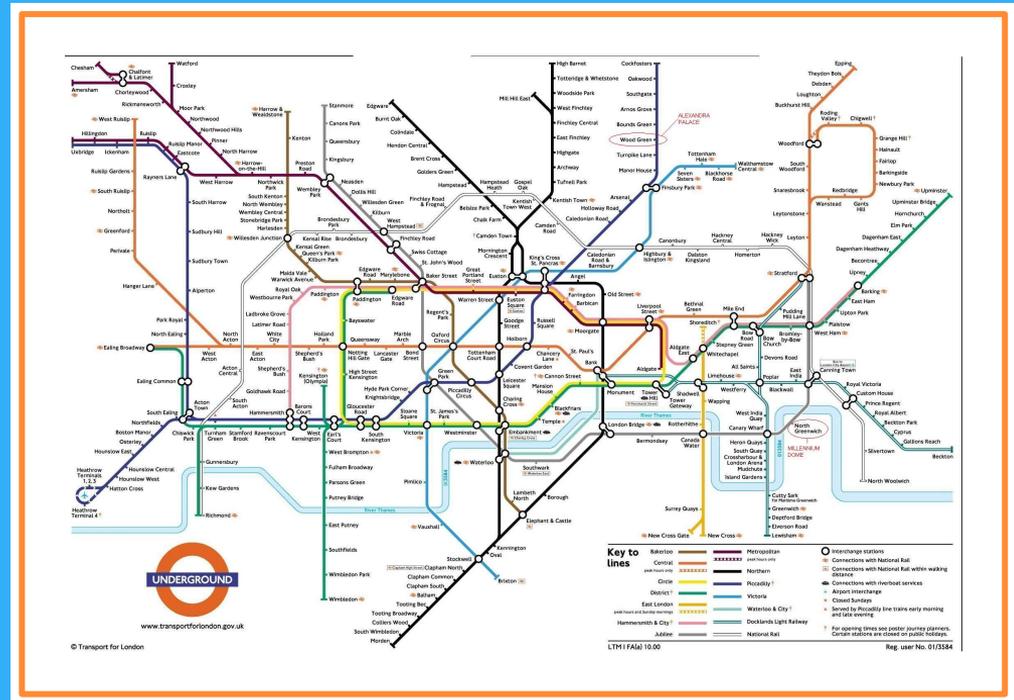


Abstraction

Réduire à un ensemble de caractéristiques essentielles

Résumer

Modèles non réalistes



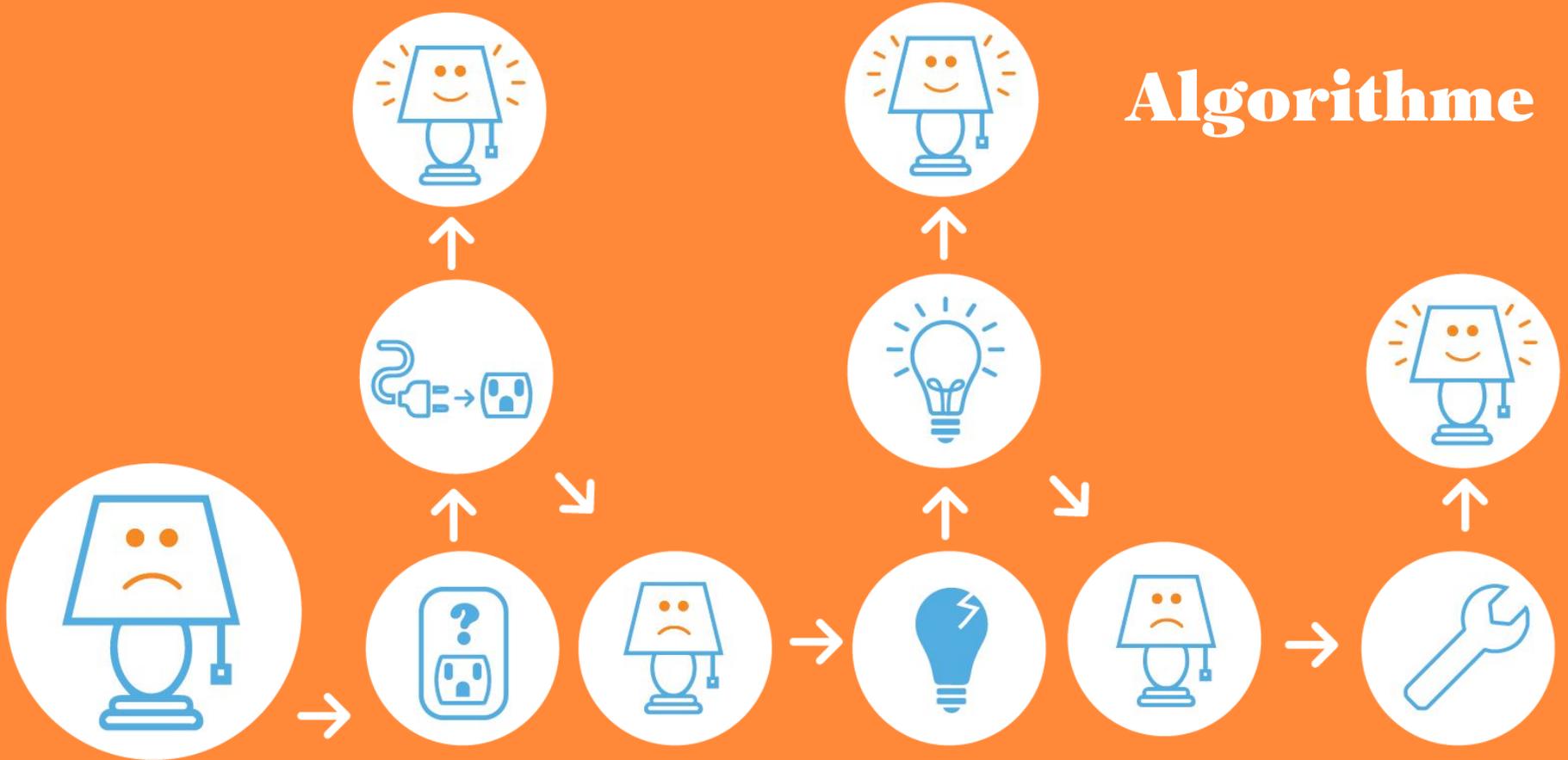
Algorithme

Une série ou séquence d'instructions étape par étape.

- 1.
- 2.
- 3.



Algorithmme



Défi de pensée informatique:

Si aujourd'hui est vendredi,
quel est le jour après le jour
avant le jour avant demain?

$\wedge - \cup /$
 $() \{ \} < >$

Logique

Analyser
et
Prédire

(cause & effet)



Si aujourd'hui est **vendredi**,
quel est

le jour après

le jour **avant**

le jour **avant**

demain?

^ - ∪ /
() { } < >

Mise en pratique

Décomposition,

Logique

+

Reconnaissance des Schémas



Si aujourd'hui est **vendredi**,
quel est

le jour après

le jour avant

le jour avant

demain?

^ - ∪ /
() { } < >

Mise en pratique

Décomposition,

Logique

+

Reconnaissance des
Schémas



Est-ce que ceci est un
problème simple maintenant?

Si aujourd'hui est **vendredi**,
quel est (le jour après (le
jour avant (le jour avant
(demain))))?

^ - ∪ /
() { } < >

Mise en pratique

Décomposition,

Logique

and

Reconnaissance des Schémas



Si aujourd'hui est **vendredi**,
quel est le jour après le jour
avant le jour avant **demain**?

^ - ∪ /
() { } < >

La pensée informatique en action!

- ✓ Décomposition
- ✓ Logique
- ✓ Reconnaissance de schémas
- ✓ Algorithme



Deuxième défi de pensée
informatique:

Additionnez
tous les chiffres
de 1 à 200,
en 10 seconds,
dans votre tête.

^ - ∪ /
() { } < >

Mise en pratique

Décomposition,

Logique

+

Reconnaissance des
Schémas



C'est évident qu'on devra
utiliser un algorithme autre
que

$$1 + 2 + 3 + \dots + 198 + 199 + 200$$

pour relever le défi avec
ses requis.

^ - ∪ /
() { } < >

Est-ce que quelqu'un
reconnait un modèle ou
séquence qui aiderait
à définir un nouvel
algorithme?



Si on regardait le problème
sous un autre angle...

$$1 + 200 = 201$$

$$2 + 199 = 201$$

...

$$100 + 101 = 201$$

Alors nous avons 100 paires,
chacune s'additionnant à 201...

^ - ∪ /
() { } < >

La pensée informatique au travail...

- ✓ Décomposition
- ✓ Logique
- ✓ Reconnaissance de schémas
- ✓ Algorithme



C'est pas du code!

Non, sérieusement



L'une des idées fausses les plus répandues à propos de la pensée informatique est que c'est la même chose que le codage. Le codage est un moyen avancé d'utiliser des concepts de pensée informatique. Le codage est la façon avec laquelle nous automatisons les abstractions et les algorithmes.

Programmer



Décomposer une solution en étapes simples,
afin que même un ordinateur puisse le faire.

Cependant...

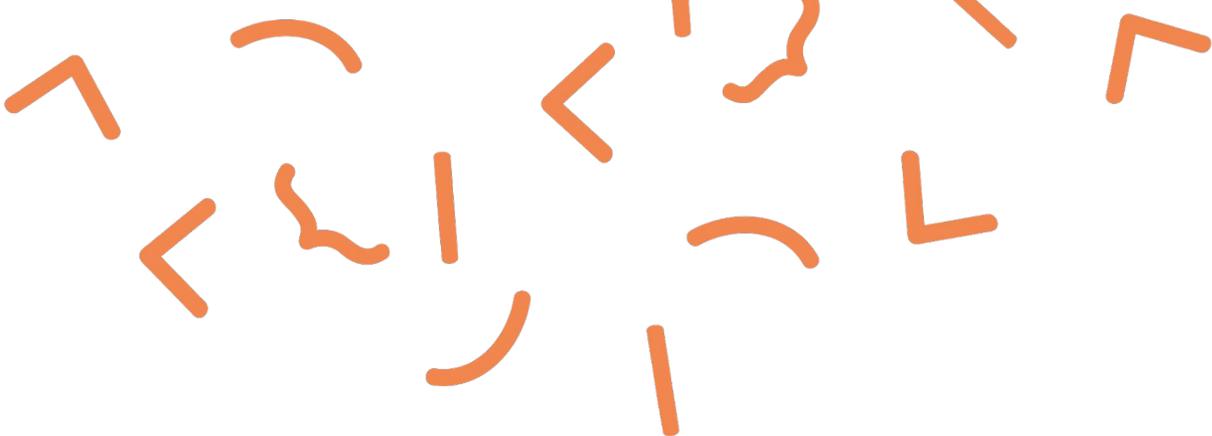


Le mouvement pour inclure la pensée informatique dans l'enseignement ne consiste pas à créer des codeurs ou des programmeurs. L'objectif est d'améliorer les compétences de résolution de problèmes d'un(e) élève en lui donnant des outils supplémentaires à utiliser lorsqu'il/elle aborde un défi. Nous ne construisons pas une génération de "bébés-programmeurs".

Pause

Twitter: @lighthouse_labs | @kidscoding

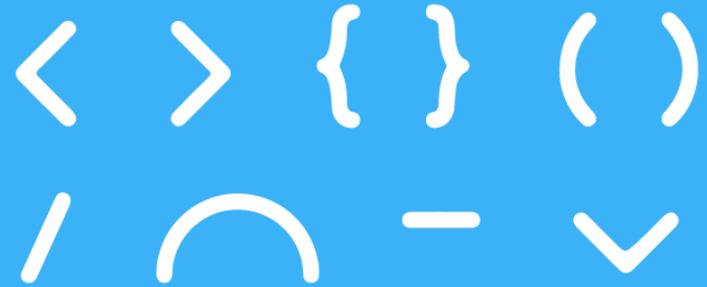
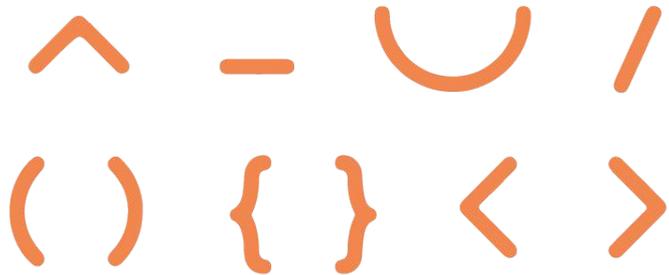
#CanCode #teacherscode

A collection of various orange geometric shapes including chevrons, arcs, and L-shapes scattered across the top left of the slide.

Activité débranchée: Jeu de Construction



Comment enseigner quand nous ne connaissons pas les réponses?



Appliquons une réflexion computationnelle à cette question ...

... en la réduisant en plus petites composantes



Qu'enseignons-nous
quand nous ne
connaissons pas la
réponse?

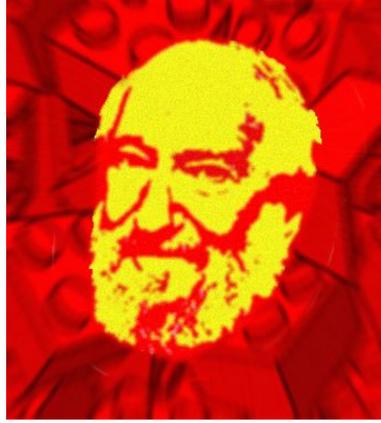
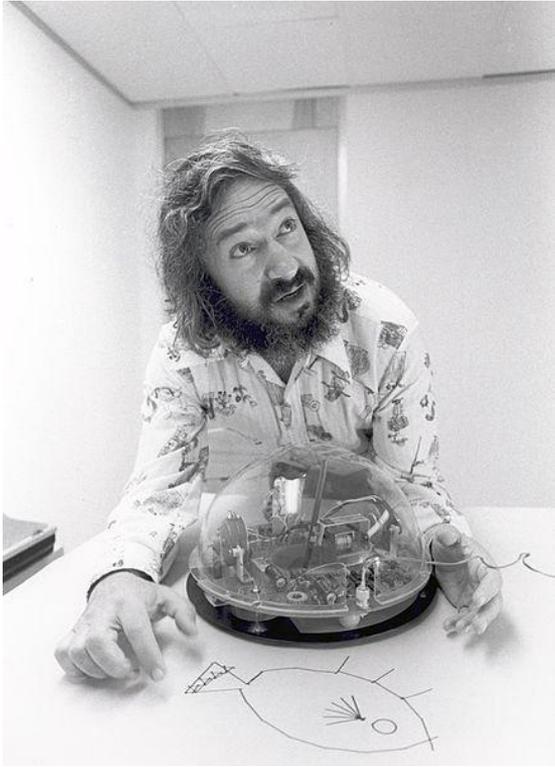


Pourquoi
voudrions-nous
enseigner de cette
façon?



Est-ce que quelqu'un a
déjà enseigné de cette
façon?





Le rôle de l'enseignant est de créer les conditions de l'inventivité plutôt que de fournir des réponses toutes faites.

Seymour Papert

Apprendre à coder permet de développer des outils cognitifs

Séquencement

Répétition

Variables

Sélection

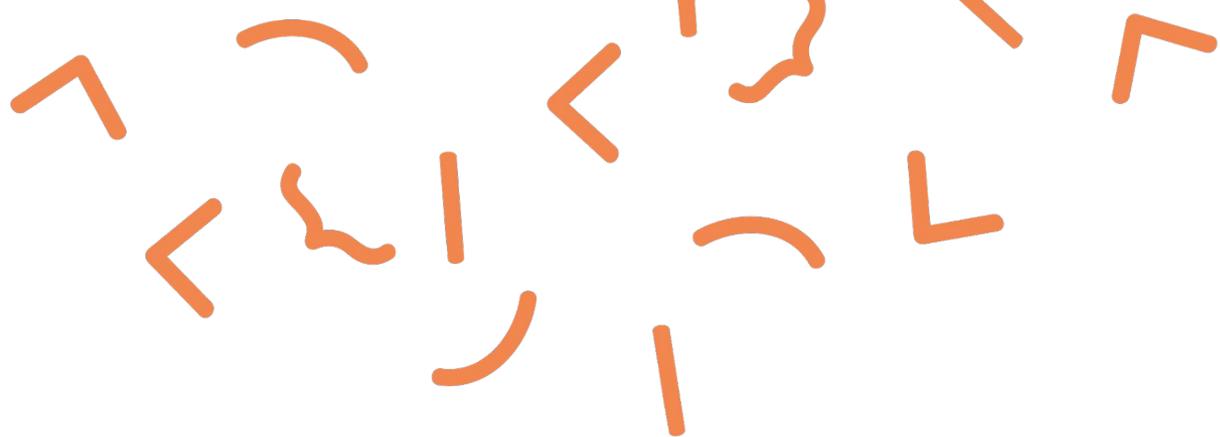
Coder pour apprendre en utilisant de nouveaux outils

Qu'est ce que la programmation
peut nous apprendre de
l'apprentissage ?

Qu'est ce que la programmation
peut nous apprendre de
l'enseignement ?

Qu'est ce que la programmation
peut nous apprendre du
raisonnement ?

Qu'est ce que la programmation
peut nous apprendre du
comportement ?



Activité débranchée:

Tour de Hanoi



SÉQUENCEMENT



Les ordinateurs sont puissants, mais ils ne sont pas très intelligents. Ils feront seulement ce qu'on leur demande, dans l'ordre qu'on leur demandera.

Pour que les ordinateurs complètent des tâches, il faut leur donner un ensemble d'instructions dans le bon ordre. C'est ce qu'on appelle le séquençement.

Il est possible d'utiliser le séquençement pour créer des programmes qui indiqueront aux ordinateurs comment faire des choses ingénieuses.

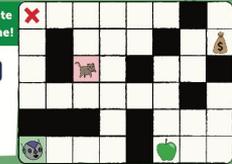


Nous utilisons le séquençement tous les jours!

Par exemple, si tu te fais une tasse de thé, tu suivras une séquence d'étapes. Tu dois compléter les étapes dans le bon ordre. Tu ne peux pas ajouter de l'eau dans la tasse avant de l'avoir fait bouillir. Peux-tu trouver d'autres exemples?

À ton tour maintenant! Expérimente le séquençement avec ce labyrinthe!

Peux-tu écrire une séquence en utilisant ces instructions afin de déplacer le robot jusqu'au chat? Tu trouveras plus de labyrinthes de ce genre dans Scratch.



Tu désires en apprendre plus à propos de Code Club? Visite www.codeclub.ca

SÉLECTION



Lorsque tu demandes au programme informatique de respecter une condition, il teste la première et si elle est fautive, il exécute la seconde.

Tu peux utiliser les commandes si et sinon afin de prendre des décisions dans tes programmes. Par exemple, lorsque tu entres ton mot de passe dans un ordinateur, il décide s'il t'en permet l'accès. C'est ce qu'on appelle la sélection.



Nous utilisons la sélection tous les jours!



Lorsque tu regardes par la fenêtre le matin, s'il pleut, tu prendras un parapluie, sinon tu laisseras le parapluie à la maison.

À ton tour maintenant! Expérimente la sélection avec ce quiz!

La sélection est aussi utilisée lorsque tu crées des quiz! Tu trouveras un programme de quiz qui utilise la sélection dans Scratch. Peux-tu créer tes propres questions de quiz?



Tu désires en apprendre plus à propos de Code Club? Visite www.codeclub.ca

RÉPÉTITION



Lorsque tu repères des groupes d'instructions qui se répètent dans ton code, tu peux utiliser la répétition au lieu de les réécrire.



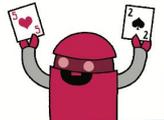
La répétition est une manière d'indiquer à l'ordinateur de répéter des instructions soit :

Un nombre de fois défini Jusqu'à temps qu'une condition soit remplie ou Indéfiniment

Par exemple, un jeu vidéo peut avoir un fond sonore en boucle jusqu'à temps que le joueur quitte le jeu.

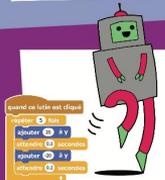
Nous utilisons la répétition tous les jours!

Si tu partages un jeu de cartes avec un ami, tu lui donnes une carte et ensuite tu en prends une. Tu répètes cela jusqu'à ce qu'il n'y en ait plus.



À ton tour maintenant! Expérimente la répétition pour faire des animations!

La répétition est très utile lorsque tu veux faire marcher, sauter ou voler un personnage. Tu peux créer des animations en utilisant la répétition dans Scratch.



Tu désires en apprendre plus à propos de Code Club? Visite www.codeclub.ca

VARIABLE



Une variable permet d'enregistrer des données avec un numéro ou du texte dans la mémoire d'un ordinateur. Il faut donner un nom à chaque variable du programme afin que les données conservées puissent être récupérées, utilisées et modifiées.



Un jeu vidéo utilise des variables afin de sauvegarder les données du jeu, comme par exemple, le pointage, le nombre de vies restantes et le temps qu'il reste à jouer.

Ces variables sont mises à jour au fur et à mesure que le jeu avance. Par exemple, le pointage augmentera pendant que le temps de jeu diminuera.

Nous utilisons des variables tous les jours!

Lorsque tu joues à des jeux de société, tu écris parfois le pointage des joueurs. Tout au long du jeu, tu changes le pointage au fur et à mesure que le jeu avance.



quand ce lutin est cliqué



À ton tour maintenant! Expérimente les variables avec ce logiciel!

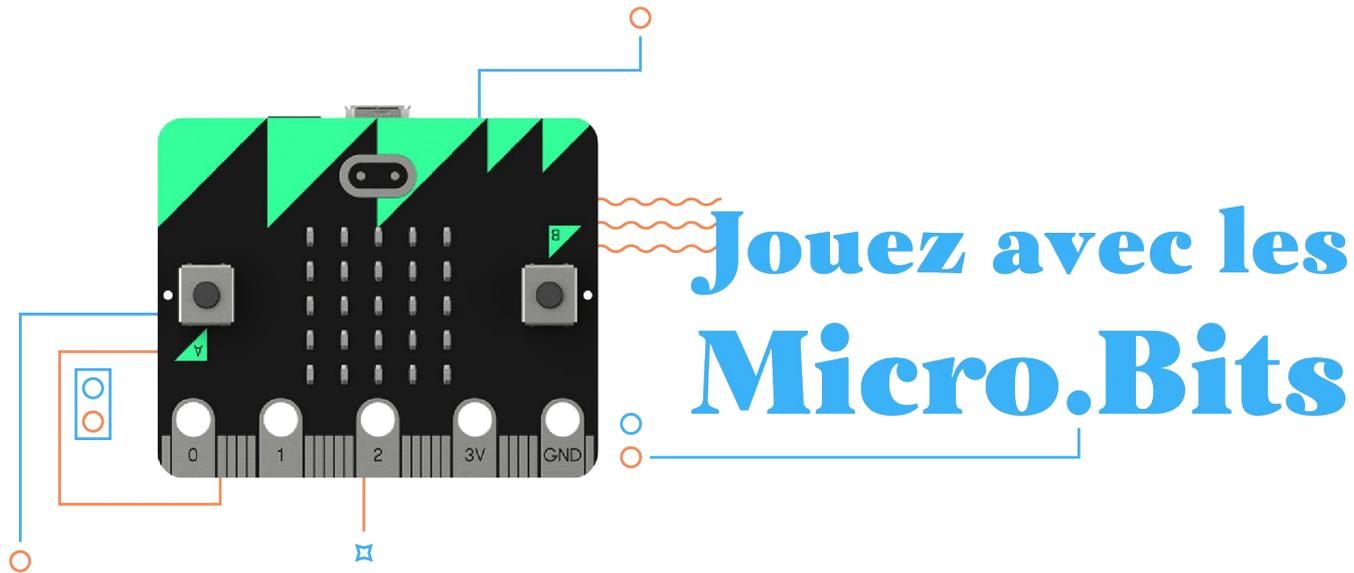
Les logiciels de vote fonctionnent en ajoutant un à la valeur enregistrée dans la variable chaque fois que tu cliques pour voter. Tu peux voter pour ton aliment préféré dans Scratch. Peux-tu ajouter des aliments pour lesquels on pourra voter?



Tu désires en apprendre plus à propos de Code Club? Visite www.codeclub.ca



Dîner



Jouez avec les Micro:Bits

Commentaires et
réflexions

réflexions

Commentaires et

Sondage:

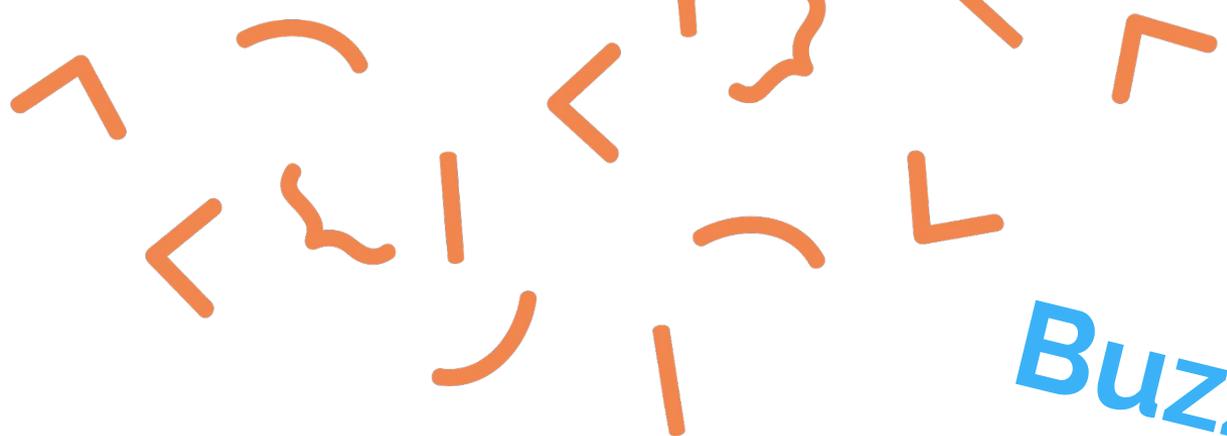
http://bit.ly/KCJ_CPFPP



Merci!

Twitter: @lighthouse_labs | @kidscoding

#CanCode



Fizz: Divisible par 3

Buzz: Divisible par 5

Fizz-Buzz.....

