

Explorateurs de cellules		8 ^e année
Plan de leçon	Outil de codage	Scratch
	Temps nécessaire	De 2 à 3 périodes
Connexions du programme de mathématiques <u>Algèbre</u> Attentes générales C3. Résoudre des problèmes et créer des représentations informatiques de situations mathématiques à l'aide de concepts et de compétences de codage. Attentes précises C3.1 Résoudre les problèmes et créer des représentations informatiques de situations mathématiques en écrivant et en exécutant du code, y compris du code qui concerne l'analyse des données afin d'informer et de communiquer les décisions. C3.2 Lire et modifier le code existant qui comprend l'analyse des données afin d'informer et de communiquer les décisions, et décrire comment les changements apportés au code influent sur les résultats et l'efficacité du code.	Connexions du programme de sciences <u>8^e année</u> Cellules <ul style="list-style-type: none"> Les cellules sont la base de la vie. Attentes générales 3. Démontrer une compréhension de la structure et de la fonction de base des cellules végétales et animales et des processus cellulaires. Attentes précises 3.1 Identifier les structures et les organites dans les cellules, y compris le noyau, la membrane cellulaire, la paroi cellulaire, les chloroplastes, la vacuole, les mitochondries et le cytoplasme, et expliquer les fonctions de base de chacune. 3.3 Comparer la structure et la fonction des cellules végétales et animales.	
Description Grâce à une série d'activités pratiques et de codage, les élèves étudieront les organites cellulaires et leurs fonctions. Les élèves apprendront que de nombreux processus cellulaires sont régis par des règles ou des modèles qui peuvent être interprétés et exprimés en code comme des énoncés conditionnels et des boucles. Ils illustreront cet apprentissage en créant un jeu Scratch jouable basé sur ces modèles dans les fonctions cellulaires.		
Critères de réussite <ul style="list-style-type: none"> Les élèves de 8^e année pourront identifier les organites trouvés dans les cellules animales et décrire leur fonction de base. Les élèves de 8^e année pourront décrire des algorithmes à l'aide de pseudocode et de codage par blocs, y compris les boucles et les instructions conditionnelles. Ils pourront démontrer comment le code peut être utilisé 	Documents et médias <ul style="list-style-type: none"> Ordinateur ou tablette avec accès à Scratch (version navigateur ou application) Un grand nombre d'objets triables de taille similaire qui peuvent être facilement triés en 3 groupes (p. ex. boules colorées ou pompons) Guide de codage de correspondance des bases nucléiques 	

pour exécuter des décisions à l'aide d'énoncés conditionnels.

- Guide de codage végétal ou animal
- Document d'algorithme de cellule

Aptitudes en réflexion computationnelle

Cette leçon aborde le code en termes de décomposition, ou de briser des problèmes complexes en parties plus petites pour le rendre plus facile à traiter. On accorde aussi une attention particulière aux énoncés conditionnels. Les instructions conditionnelles sont utilisées dans le codage pour exécuter une condition si une instruction est vraie (cela peut être considéré comme semblable à aider un ordinateur à prendre une décision programmée en fonction de facteurs ou renseignements donnés).

Dans l'activité non branchée, les élèves sont encouragés à considérer les fonctions des organites en termes d'énoncés conditionnels (si/alors) et pratiquer à écrire les « règles » qui régissent ces fonctions comme pseudocode. Il s'agit d'une étape utile pour comprendre comment les langages de codage, en particulier les éléments d'imbrication communs aux boucles et aux énoncés conditionnels, s'intègrent et d'un excellent outil à utiliser lorsque les élèves planifient le code.

Dans l'activité de codage en ligne, les élèves utiliseront Scratch pour créer et modifier un jeu qui utilise des variables, des boucles et des instructions conditionnelles pour transformer les entrées (les presse-clés du joueur) en augmentations de score pour les choix corrects, et diminutions de score pour les choix incorrects. Le sujet du jeu renforce l'apprentissage des élèves au sujet des cellules (avec des versions alternatives décrites pour convenir à votre classe).

Les documents guide de programmation de **correspondance des bases nucléiques** et guide de codage **végétal ou animal** pour cette leçon comprennent une procédure détaillée étape par étape pour l'utilisation de Scratch.

Introduction

Introduction aux cellules et aux organites

Les cellules sont l'unité de vie de base. Ils contiennent des parties plus petites, appelées organites, qui sont spécialisées pour effectuer des tâches spécifiques.

Commencez par examiner les organites trouvés dans les cellules animales et leurs fonctions de base.

- Membrane cellulaire : contrôle de ce qui entre et sort de la cellule
- Cytoplasme : n'est pas techniquement un organite, mais il contient/suspend des organites à l'intérieur de la cellule
- Ribosomes
- Réticulum endoplasmique : se présente sous forme lisse (SER) et (RER). SER produit des lipides (graisses) qui forment la membrane cellulaire; RER soutient la production de protéines (pour les protéines quittant la cellule)

- Mitochondrie : production énergétique
- Ribosome : construire des protéines en décodant l'information (à partir de l'ARN) et en plaçant des acides aminés (les blocs de base des protéines) dans le bon ordre selon l'information décodée.
- Appareil de Golgi : modification et emballage des protéines
- Lysosome : destruction des protéines
- Vacuole : stockage des déchets, de l'eau ou des nutriments (selon le type de cellule)
- Noyau : Stockage de l'ADN

Cette leçon examinera comment nous pouvons décrire les fonctions des organites en fonction d'un ensemble de règles, ou d'un algorithme. Nous allons examiner de plus près la membrane cellulaire, les ribosomes et le noyau.

Introduction aux énoncés conditionnels

Nous utilisons des énoncés conditionnels tous les jours sans y penser. Un énoncé conditionnel peut être considéré comme une décision fondée sur la réponse à une question. Pensez à la décision de porter vos mitaines.

La question à laquelle vous répondez pourrait être :

« Fait-il froid dehors? » ou « mes mains ont-elles froid? » ou « mes mains seront-elles froides si je sors? »

Disons que la question à laquelle nous répondons pour décider de porter des mitaines est « mes mains ont-elles froid? »

On peut répondre à cette question par un oui (mes mains sont froides) ou par un non (mes mains ne sont pas froides).

Dans la vraie vie, vous déciderez probablement de mettre vos mitaines sur vos mains si vous avez froid et de ne pas mettre vos mitaines sur vos mains si vous n'avez pas froid.

Dans le code, cette décision peut être représentée sous la forme d'un énoncé conditionnel, appelé énoncé si/alors :

Si mes mains sont froides, **alors** je mets mes mitaines.

Dans ce cas, la partie **si** de l'énoncé est la condition qui est remplie (les mains sont froides) et **alors** est le programme qui est exécuté parce que la condition est satisfaite. Nous pouvons même programmer la décision de laisser vos mitaines hors de vos mains parce que vos mains

ne sont pas froides avec un énoncé **sinon**. Il pourrait être utile de penser à **sinon** comme « autrement ».

Si mes mains sont froides, **alors** je mets mes mitaines. **Sinon**, je ne mets pas mes mitaines.

Nous pouvons également avoir plus d'une condition qui peut conduire à une décision. Par exemple, nous avons soulevé quelques questions qui pourraient nous amener à mettre des mitaines. Nous pourrions ajouter une règle pour décrire que nous mettrions des mitaines en cas de froid, même si nos mains ne sont pas froides (encore) en utilisant un énoncé **sinon, si**. Il pourrait être utile de penser à **sinon, si** comme « OU si ».

Si mes mains sont froides, **sinon s'il** fait froid, **alors** je mets mes mitaines. **Sinon**, je ne mets pas mes mitaines.

La façon dont nous décrivons ces conditions en langage clair est une forme de pseudocode. Pseudocode est un outil utile pour planifier un programme de codage avant de passer à un langage de codage.

Nous allons utiliser un pseudocode pour examiner les décisions que différents organites utilisent pour effectuer leurs tâches à l'intérieur des cellules animales.

Mesure

Activité débranchée

La membrane cellulaire protège la cellule et contrôle les molécules qui entrent dans la cellule (comme les nutriments) et qui sortent de la cellule (comme les déchets). Certaines de ces molécules traversent facilement les ouvertures dans la membrane (transport passif, comme diffusion ou osmose). D'autres molécules doivent être jumelées à une molécule porteuse pour obtenir de l'aide pour se déplacer à travers la membrane (transport actif).

Utilisez la partie 1 du **Document sur les algorithmes des cellules** pour définir les règles de l'activité de la membrane cellulaire en tant qu'expressions conditionnelles. Ceci sera écrit sous forme de pseudocode, par exemple :

Le but de notre programme de membrane cellulaire est de vérifier si une boule est bleue ou rouge ou jaune*. Il laissera passer des boules rouges individuelles, mais pas des boules bleues ou jaunes individuelles. Les boules bleues et jaunes ne peuvent passer que si elles sont jumelées (une boule bleue + une boule jaune ensemble).

Donc, comme pseudocode :

si la balle est rouge

alors elle passe à travers la membrane

sinon, si la balle est (une paire) bleue ET jaune

alors elle passe à travers la membrane

sinon

alors elle ne passe pas à travers la membrane

*Les objets que votre classe est en train de trier et les critères de tri peuvent être ce que vous voulez.

Maintenant, essayez d'exécuter les règles pour déplacer les balles d'un côté de la classe à l'autre. Divisez les élèves en trois groupes :

- 1) Le groupe 1 (à l'extérieur de la cellule) recueillera des balles (une balle par élève à la fois) pour les amener à la membrane cellulaire.
- 2) Le groupe 2 agira comme la membrane cellulaire. Ils passeront immédiatement des balles rouges aux élèves du groupe 3 (à l'intérieur de la cellule) et tiendront des balles bleues ou jaunes jusqu'à ce qu'ils aient une paire (1 boule jaune + 1 boule bleue). Une fois qu'ils ont une paire, ils passeront les deux balles aux élèves du groupe 3.
- 3) Le groupe 3 (à l'intérieur de la cellule) récupère les balles de la membrane cellulaire et les place dans une poubelle (max 1 boule rouge ou 1 paire jaune-bleu à la fois).

Possibilité d'extension : nous pouvons ajouter des règles supplémentaires au jeu pour le rendre plus difficile. Par exemple : les étudiants qui agissent comme membrane cellulaire ne peuvent tenir qu'un maximum de deux balles à la fois. Comment exprimeriez-vous cela comme un énoncé conditionnel dans un pseudocode?

Exemple :

Pseudocode	Explication
répéter jusqu'à Valeurballe = 2 recevoir une balle; pour chaque balle reçue, augmenter [Valeurballe] + 1;	Il s'agit d'une boucle avec un compteur. Chaque fois qu'un étudiant reçoit une balle, la variable Valeurballe augmente de +1. Une fois que la variable Valeurballe est égale à 2, le code sort de la boucle et passe à la section suivante du code.
vérifier Couleurballe si Couleurballe ne correspond pas	Il s'agit d'un énoncé conditionnel . L'élève vérifie les deux balles qu'il tient pour s'assurer que la paire ne

<p>passer les balles à travers la membrane;</p> <p>sinon</p> <p>ne pas passer les balles.</p>	<p>correspond pas (p. ex. jaune et bleu). Si les couleurs ne correspondent pas, le si de la condition de l'énoncé est VRAI et les balles peuvent passer la membrane selon les règles du programme. Si les couleurs correspondent, le si de la condition de l'énoncé est FAUX et donc le programme exécutera à la place la sortie pour la condition sinon, qui est de ne pas permettre aux balles de passer la membrane.</p>
--	--

Complétez le document algorithme des cellules pour plus d'exercices pratiques en décomposant les étapes des fonctions cellulaires sous forme de pseudocode. Le deuxième exemple de la distribution porte spécifiquement sur les ribosomes et la façon dont ils construisent des chaînes d'acides aminés selon des groupements à trois lettres appelés codons. Chaque lettre d'un codon représente une base nucléique, qui est la « langue » avec laquelle nous encodons et stockons l'information dans notre ADN. Il y a quatre bases dans l'ADN : A (adénine), C (cytosine), G (guanine), T (thymine) – nous nous référerons à ces bases par leurs lettres de cette leçon.

Les ribosomes lisent une forme transcrite d'ADN appelée ARNm pour construire des protéines à partir de blocs de construction appelés acides aminés. L'ARNm utilise aussi quatre bases, tout comme l'ADN. La seule différence est que T (thymine) est remplacé par une nouvelle base U (Uracil). Donc, les quatre bases de l'ARNm sont : A, C, G et U.

Ces bases peuvent faire 64 combinaisons différentes de groupements de trois bases pour coder pour 20 types d'acides aminés. Pour simplifier les choses, notre document ne nous montre que quatre de ces combinaisons (appelées codons), ainsi que leur acide aminé associé (et une forme) :

- UUA = Phe (triangle)
- AUG = Met (cercle)
- CGC = Arg (étoile)
- UCU = Ser (carré)

Traduire la chaîne suivante de 10 codons, quelle serait la séquence résultante d'acides aminés? Quelle serait la séquence résultante de formes?

AUGCGCCGCUCUUUAUCUUUACGCUUACGC

Réponse

Met-Arg-Arg-Ser-Phe-Ser-Phe-Arg-Phe-Arg

OU

Cercle – Étoile – Étoile – Carré – Triangle – Carré – Triangle – Étoile – Triangle – Étoile

Quel ensemble de règles avez-vous utilisé pour traduire les codons en acides aminés?

Comment écririez-vous ceci en pseudocode?

Exemple :

Répétez jusqu'à ce que toutes les bases nucléiques dans l'ordre soient codées :

Lire les bases nucléiques en groupes de trois à partir de la première base à gauche;

si le regroupement est AUG

alors placez Met (ou cercle);

sinon, si le regroupement est CGC

alors placez Arg (ou étoile);

sinon, si le regroupement est UCU

alors placez Ser (ou carré);

sinon

placez Phe (ou triangle).

Activité de codage

Rappelez-vous que l'ADN est stocké dans le noyau de la cellule et que l'ADN est « écrit » en quatre lettres, ou bases : A, T, G et C. Si vous imaginez les molécules d'ADN comme une échelle, chaque échelon de l'échelle est en fait deux de ces bases combinées. A est toujours jumelé avec T. G est toujours associé à C.

Nous allons utiliser **Scratch** pour créer un jeu qui utilise les mêmes règles de correspondance pour les bases d'ADN (voir note ci-dessous pour une version alternative de ce jeu qui peut être construit).

Voici ce que nous voulons que notre jeu fasse :

- Chaque lettre A, T, G et C sera affectée à une touche fléchée sur un clavier (ou à un bouton tactile sur iPad);

- Le jeu affichera aléatoirement une lettre (A, T, G ou C) à la fois sur l'écran;
- Le joueur appuie sur la touche fléchée de la lettre qui est **jumelée** à la lettre à l'écran (si l'écran affiche A, alors le joueur doit appuyer sur la flèche attribuée à T);
- Si le joueur devine correctement, le score augmente d'un point;
- Si le joueur est incorrect, le score diminue d'un point.

Un guide détaillé étape par étape pour construire ce jeu est décrit dans le **Guide de codage de correspondance des bases nucléiques**.

Un exemple de version de ce code peut être visualisé et lu ici :

<https://scratch.mit.edu/projects/438699096/editor/>

Si nous pensons en termes de pseudocode comme nous l'avons fait avec les activités précédentes, l'algorithme du jeu se décompose comme suit :

Pseudocode	Explication
Répéter pour toujours	Il s'agit d'une boucle qui contiendra le programme. Nous voulons que notre jeu vérifie et réponde aux clés pressées par le joueur indéfiniment.
si la lettre A est à l'écran, alors	Il s'agit d'un énoncé conditionnel . Parce que nous avons quatre programmes en cours d'exécution simultanée et la vérification en boucle pour voir quel costume de lettre est affiché, nous avons besoin d'une condition pour déclencher ce programme. Lorsque A est affiché (et non T, C ou G), cette condition est remplie et le programme pour A est exécuté (et non le programme pour T, C ou G).
si la flèche droite est enfoncée alors modifier le score par +1	Ceci est un énoncé conditionnel (imbriqué dans le précédent!). Si le joueur appuie sur la touche flèche droite (la touche attribuée à T, ou la base qui correspond à A), la touche correcte a été enfoncée et le compteur de score augmentera la valeur de score de 1.
si une flèche vers le haut, une flèche vers le bas ou une flèche vers la gauche sont enfoncées alors modifier le score par -1	Il s'agit d'un autre énoncé conditionnel (imbriqué au même niveau que l'énoncé précédent).

La touche flèche droite est la seule touche correcte, donc si le joueur appuie sur une autre touche, une touche incorrecte a été enfoncée et le compteur de score diminuera la valeur de score de 1.

Autre version : Cellule végétale ou animale?

Ce jeu peut être facilement réécrit pour être un outil d'étude pour l'apprentissage des organites et identifier lesquels peuvent être trouvés dans les cellules végétales, les cellules animales, ou les deux.

Voici ce que nous voulons que notre jeu fasse :

- Trois options (cellule végétale, cellule animale, les deux) seront assignées aux touches fléchées du clavier (ou boutons tactiles sur iPad);
- Le jeu affichera aléatoirement un nom d'organite à la fois sur l'écran;
- Le joueur appuie sur la touche de flèche pour le type de cellule qui correspond à cet organite (végétale, animale, ou les deux);
- Si le joueur devine correctement, le score augmente d'un point;
- Si le joueur est incorrect, le score diminue d'un point.

Cette version est par ailleurs construite de la même manière que le jeu de correspondance des bases nucléiques. Il code pour moins de clés jouables (3 au lieu de 4), mais a plus de code dans l'ensemble (un programme pour chaque organite que vous incluez dans le jeu).

L'exemple de code est inclus dans le **Guide de codage végétal ou animal**.

Clôture et évaluation

À la fin de cette leçon, les élèves devraient être en mesure de reconnaître les organites des cellules animales et de décrire leurs fonctions de base. Les élèves devraient être en mesure de modifier les énoncés conditionnels pour modifier la sortie d'un programme créé numériquement, de lire leur code et d'itérer.

Pour l'évaluation, recueillez les documents « algorithmes des cellules » des étudiants. Revoyez leur travail pour vous assurer qu'ils comprennent les concepts d'algorithmes et d'énoncés conditionnels en décrivant de manière appropriée les étapes nécessaires pour que les membranes cellulaires et les ribosomes exécutent leurs tâches.

<p>Adaptations</p> <ul style="list-style-type: none">• L'activité de tri des membranes cellulaires hors ligne peut être effectuée assis en classe ou assis individuellement à leur bureau.• Les objets utilisés pour l'activité de tri des membranes cellulaires peuvent être effectués à l'aide d'objets triés par des caractéristiques autres que la couleur (comme la forme ou le type).	<p>Extensions</p> <ul style="list-style-type: none">• Les étudiants qui finissent tôt peuvent apporter des modifications à leur code pour voir comment cela affecte la jouabilité. Les modifications peuvent inclure l'ajustement du défi du jeu en changeant la valeur dans le bloc d'attente ou les valeurs de score pour différentes conditions ou en ajustant l'esthétique du jeu en changeant les sons joués ou l'apparence du jeu.
<p>Ressources supplémentaires</p> <ul style="list-style-type: none">• Scratch.mit.edu – Scratch est une ressource gratuite et aucun compte n'est nécessaire pour créer un programme; toutefois, un compte est nécessaire pour sauvegarder votre travail.	