

Initiation à la programmation : De Scratch vers la seconde

Apprendre la programmation permet de réaliser très rapidement des projets utiles ou ludiques (jeux-vidéos, retouche photos, applications ...). Cette activité n'est pas vraiment difficile, mais il faut de la méthode et de la persévérance.

Faire de la programmation consiste à écrire un ensemble d'instructions, qui s'enchaînent les unes après les autres, dans un ordre déterminé, de manière à atteindre un (ou des) objectif(s) fixés au préalable. C'est ce que nous appelons un algorithme.

Dans un algorithme :

- Une instruction est une série d'actions décrites à l'aide de mot clés et dont l'enchaînement se présente grâce aux connecteurs logiques. Une instruction contient très souvent une variable.
- Une variable peut être assimilée à une boîte, désignée par un nom, et dont le contenu peut évoluer au fil de l'évolution de l'algorithme.
- Le langage utilisé pour l'écrire peut être le « langage naturel » (structure logique écrite en français) ou un langage informatique (structure logique écrite dans un langage que les ordinateurs peuvent interpréter).

Un algorithme se décompose (pour l'instant) en différentes étapes répondant chacune à une question particulière :

- Quelles sont les variables (et leur type) ?
- Quelles sont les valeurs initiales de ces variables ?
- Que saisit l'utilisateur au départ ?
- Quelles sont les instructions exécutées par l'algorithme ?
 - Affectation de valeurs
 - Instruction conditionnelle
 - Boucle bornée,
 - Boucle non bornée,
 - Récupération d'informations saisies par l'utilisateur,
 - Affichage d'informations ...
- Que veut-on afficher à la fin de l'exécution ?
- Quand souhaite-t-on arrêter l'algorithme ?

Il est fortement conseillé d'utiliser le logiciel libre et gratuit Scratch tout au long de ce module. Vous pouvez soit le télécharger sur votre ordinateur, soit l'utiliser directement en ligne à partir de l'adresse suivante :

<https://scratch.mit.edu/>

Partie n°1 : Notion de variable / Affectation / Lecture – Ecriture / Reconstituer un programme

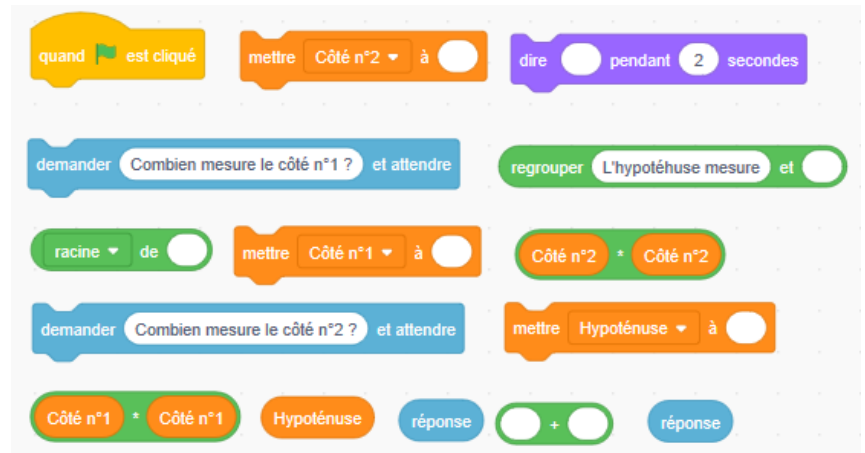
Exercice n°1 : Autour du théorème de Pythagore

Avec le langage Scratch

D'après le théorème de Pythagore, dans un triangle rectangle, le carré de l'hypoténuse est égal à la somme des carrés des deux autres côtés.

Nolan a écrit un programme qui calcule et affiche la longueur de l'hypoténuse d'un triangle rectangle à partir de la longueur des côtés de l'angle droit.

Mais Mathis son petit frère s'est amusé à les mettre dans le désordre.



1. Aider Nolan a reconstituer son programme en écrivant les instructions dans le bon ordre.

Instruction 1 : Quand drapeau est cliqué

Instruction 2 : Demander « combien mesure le côté n°1 » et attendre

Instruction 3 : Mettre Côté n°1 à réponse

Instruction 4 : Demander « combien mesure le côté n°2 » et attendre

Instruction 5 : Mettre Côté n°2 à réponse

Instruction 6 : Mettre Hypoténuse à racine de Côté n°1 * Côté n°1 + Côté n°2 * Côté n°2

Instruction 7 : Dire regrouper « l'hypoténuse mesure » et Hypoténuse

2. A quoi sert la première instruction de cet algorithme ?

La première instruction permet de lancer l'exécution du programme.

3. A quoi correspond le bloc « Côté n°1 » apparaissant dans la 3ième instruction de cet algorithme ?

Le bloc Côté n°1 correspond à une variable de type numérique

4. Quelle est la différence entre les blocs « Côté n°1 » et « réponse » ?

Côté n°1 et réponse sont tous les deux des variables. La différence est que la variable est déjà créée dans Scratch alors que Côté n°1 doit être créée.

5. Identifier toutes les variables de cet algorithme.

Les variables sont Côté n°1 ; Côté n°2 ; Hypoténuse (et réponse)

6. Comment sont initialisées ces variables ?





Ces variables sont initialisées grâce aux instructions « mettre à »

7. Qu'affiche-t-on à la fin de l'exécution ?


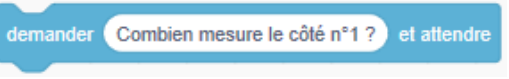

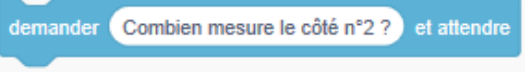


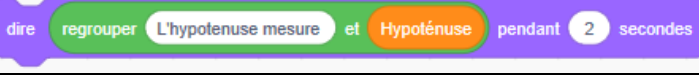
La dernière instruction affiche à la fois du texte : « l'hypoténuse mesure » et le contenu de la variable Hypoténuse.

Avec le langage naturel :

La plupart des algorithmes, avant d'être écrit dans un langage informatique, sont dans un premier temps écrit en langage naturel. Voici un tableau contenant quelques-uns des blocs rencontrés dans Scratch et leur traduction en langage naturel.

Instruction Scratch	Langage naturel
	Début algorithme <i>On note aussi « Fin algorithme » lorsque celui-ci doit s'arrêter.</i>
	Afficher le message : « » Lire Réponse
	Affecter « » à 'a' (ou 'a' prend la valeur ...)
	Afficher « phrase »

Traduire le script précédent en langage naturel en utilisant le tableau de correspondance

Instruction	Langage naturel
	Début algorithme :
	Afficher le message : Combien mesure le côté n°1 ? Lire Côté n°1
	Affecter Réponse à Côté n°1
	Afficher le message : Combien mesure le côté n°2 ? Lire Côté n°2
	Affecter Réponse à Côté n°2
	Affecter $\sqrt{Côté\ n^{\circ}1^2 + Côté\ n^{\circ}2^2}$ à Hypoténuse
	Afficher « L'hypoténuse mesure », Hypoténuse
	Fin algorithme

Exercice n°2 : Volume d'un pavé droit

Un pavé droit a pour largeur « l », pour hauteur « h » et pour profondeur « p ». Les instructions suivantes permettent de créer un programme qui calcul et affiche le volume de ce pavé droit.



1. Quelles sont les différentes variables de ce programme ?
Les variables de ce programme sont profondeur ; largeur ; hauteur et volume.
2. Quelles sont les différentes instructions d'affectations de ce programme
Les instructions d'affectations sont les instructions « mettre à »
3. Remettre les instructions dans le bon ordre.
Instruction 1 : Quand drapeau est cliqué
Instruction 2 : Demander « combien mesure la largeur ? » et attendre
Instruction 3 : Mettre largeur à réponse
Instruction 4 : Demander « combien mesure la hauteur ? » et attendre
Instruction 5 : Mettre hauteur à réponse
Instruction 6 : Demander « combien mesure la profondeur » et attendre
Instruction 7 : Mettre profondeur à réponse
Instruction 8 : Mettre volume à valeur largeur*hauteur*profondeur
Instruction 9 : Afficher « le volume est de », volume
4. Traduire ce programme en langage naturel
Instruction 1 : Début algorithme
Instruction 2 : Afficher le message « combien mesure la largeur ? »
Instruction 3 : Lire largeur
Instruction 4 : Afficher le message « combien mesure la hauteur ? »
Instruction 5 : Lire hauteur
Instruction 6 : Afficher le message « combien mesure la profondeur ? »
Instruction 7 : Lire profondeur
Instruction 8 : volume prend la valeur largeur*hauteur*profondeur
Instruction 9 : Afficher le message « le volume est de », volume
Instruction 1 : Fin algorithme

Exercice n°3 : Volume d'une pyramide régulière

Une pyramide régulière a pour hauteur « h » et pour base un carré de côté « a ». Les variables « h » et « a » sont exprimés dans une même unité de longueur.

On souhaite créer un programme en langage naturel puis en langage Scratch permettant de calculer le volume de cette pyramide. Voici l'algorithme incomplet.

Instructions	Langage naturel	Langage Scratch
1	Début Algorithme	Quand drapeau est cliqué
2	Demander « Quelle est la longueur du côté de cette pyramide ? »	Demander « Quelle est la longueur du côté de cette pyramide ? » et attendre
3	Lire a	Mettre a à réponse
4	Demander « Quelle est la hauteur de cette pyramide ? »	Demander « Quelle est la hauteur de cette pyramide ? » et attendre
5	Lire h	Mettre h à réponse
6	Base prend la valeur a^2	Mettre Base à réponse
7	Volume prend la valeur $\frac{Base \times h}{3}$	Mettre Volume à Base * h / 3
8	Afficher « le volume est de », Volume	Dire regrouper « le volume est de » et volume
9	Fin algorithme	

- Quelles sont les différentes variables nécessaires pour faire fonctionner ce programme ?
 a : Variable correspondant à la longueur du côté de la base (*type numérique*)
 h : Variable correspondant à la hauteur de la pyramide (*type numérique*)
 Base : Variable correspondant à la valeur de l'aire de la base (*type numérique*)
 Volume : Variable correspondant à la valeur du volume de la pyramide (*type numérique*)
- Quelles sont les différentes instructions d'affectations de ce programme
 Les instructions d'affectation de ce programme :
 - en langage naturel sont les instructions « Lire » et « ... prend ... »
 - dans le script Scratch sont les instructions « Mettre à ... »
- Compléter le programme en langage naturel et en langage Scratch.

Exercice n°4 : Un programme de calcul en algorithme

Ecrire un script correspondant au programme de calcul suivant :

Programme de calcul	Langage naturel	Langage Scratch
Choisir un nombre Ajouter 3,7 Multiplier par -3 Retraire 3,5 Afficher le résultat	Début algorithme Demander « Choisir un nombre » Lire x x prend la valeur $x + 3,7$ x prend la valeur $x \times (-3)$ x prend la valeur $x - 3,5$ Afficher « le résultat est », x Fin algorithme	Quand drapeau est cliqué Demander « Choisir un nombre » et attendre Mettre x à réponse Mettre x à Mettre x à Mettre x à Dire regroupe « le résultat est » et x

Exercice n°5 : Synthèse sur l'affectation de variables

1. Pour chaque algorithme, indiquer le résultat des instructions qui le composent.

Algorithme n°1

Début algorithme
Variables A : Entier	<i>Création de la variable A de type entier</i>
B : Entier	<i>Création de la variable B de type entier</i>
A prend la valeur 1	<i>A=1</i>
B prend la valeur A + 3	<i>B=4</i>
A prend la valeur 3	<i>A=3</i>
Fin algorithme

Algorithme n°2

Début algorithme
Variables A : Entier	<i>Création de la variable A de type entier</i>
B : Entier	<i>Création de la variable B de type entier</i>
C : Entier	<i>Création de la variable C de type entier</i>
A prend la valeur 5	<i>A=5</i>
B prend la valeur 3	<i>B=3</i>
C prend la valeur A+B	<i>C=8</i>
A prend la valeur 2	<i>A=2</i>
C prend la valeur B-A	<i>C=1</i>
Fin algorithme

Algorithme n°3

Début algorithme
Variables A : Entier	<i>Création de la variable A de type entier</i>
B : Entier	<i>Création de la variable A de type entier</i>
A prend la valeur 5	<i>A=5</i>
B prend la valeur 3	<i>B=3</i>
A prend la valeur B	<i>A=3</i>
B prend la valeur A	<i>B=3</i>
Fin algorithme

2. Pour l'algorithme n°3, les deux dernières instructions permettent-elles d'échanger les valeurs de la variable A et de la variable B ?

On constate que non, les variables A et B sont toutes les deux à la valeur 3.

3. En vous appuyant sur l'algorithme n°3, écrire un algorithme permettant d'échanger les valeurs des variables A et B.

Variable C : Entier

A prend la valeur 5

B prend la valeur 3

C prend la valeur A

A prend la valeur B

B prend la valeur C

Exercice n°6 : Synthèse sur la lecture / écriture

Pour chaque algorithme, indiquer le résultat de chaque instruction puis indiquer à quoi sert l'algorithme.

Algorithme n°1

Début algorithme
Variables : nombre : numérique	<i>Création de la variable nombre de type numérique</i>
double : numérique	<i>Création de la variable double de type numérique</i>
nombre prend la valeur 5	<i>nombre = 5</i>
double prend la valeur 2*nombre	<i>double = 10</i>
Ecrire « la valeur de départ est », valeur	<i>la valeur de départ est 5</i>
Ecrire « son double est », double	<i>son double est 10</i>
Fin algorithme

Cette algorithme permet de calculer le double d'un nombre saisi par l'utilisateur.

Algorithme n°2

Début algorithme
Variables nombre : numérique	<i>Création de la variable nombre de type numérique</i>
carré : numérique	<i>Création de la variable carré de type numérique</i>
Ecrire « Entrer un nombre »	<i>Entrer un nombre</i>
Lire nombre	<i>Affecte la valeur tapée à la variable nombre</i>
Carré prend la valeur nombre*nombre	<i>Affecte le carré du nombre à la variable Carré</i>
Ecrire « Le carré du nombre choisi est », carré	<i>Le carré du nombre est ...</i>
Fin algorithme

Cette algorithme permet de calculer le carré d'un nombre saisi par l'utilisateur.

Algorithme n°3

Début algorithme
Variables nb : numérique	<i>Création de la variable A de type numérique</i>
P_HT : numérique	<i>Création de la variable A de type numérique</i>
P_TTC : numérique	<i>Création de la variable A de type numérique</i>
T_TVA : numérique	<i>Création de la variable A de type numérique</i>
Ecrire « Entrer le prix hors taxes de l'article ? »	<i>Affiche la question</i>
Lire P_HT	<i>Enregistre la réponse dans P_HT</i>
Ecrire « Combien y-a-t-il d'articles ? »	<i>Affiche la question</i>
Lire nb	<i>Enregistre la réponse dans nb</i>
Ecrire « Quel est le taux de TVA ? »	<i>Affiche la question</i>
Lire T_TVA	<i>Enregistre la réponse dans T_TVA</i>
P_TTC prend la valeur $nb * P_HT * (1 + T_TVA)$	<i>$P_TTC = nb \times P_HT \times (1 + T_TVA)$</i>
Ecrire « Le prix TTC est », P_TTC	<i>Le prix TTC est</i>
Fin algorithme

Cette algorithme permet de calculer le prix TTC d'une commande en fonction du prix HT d'un article, du nombre d'articles et du taux de TVA.

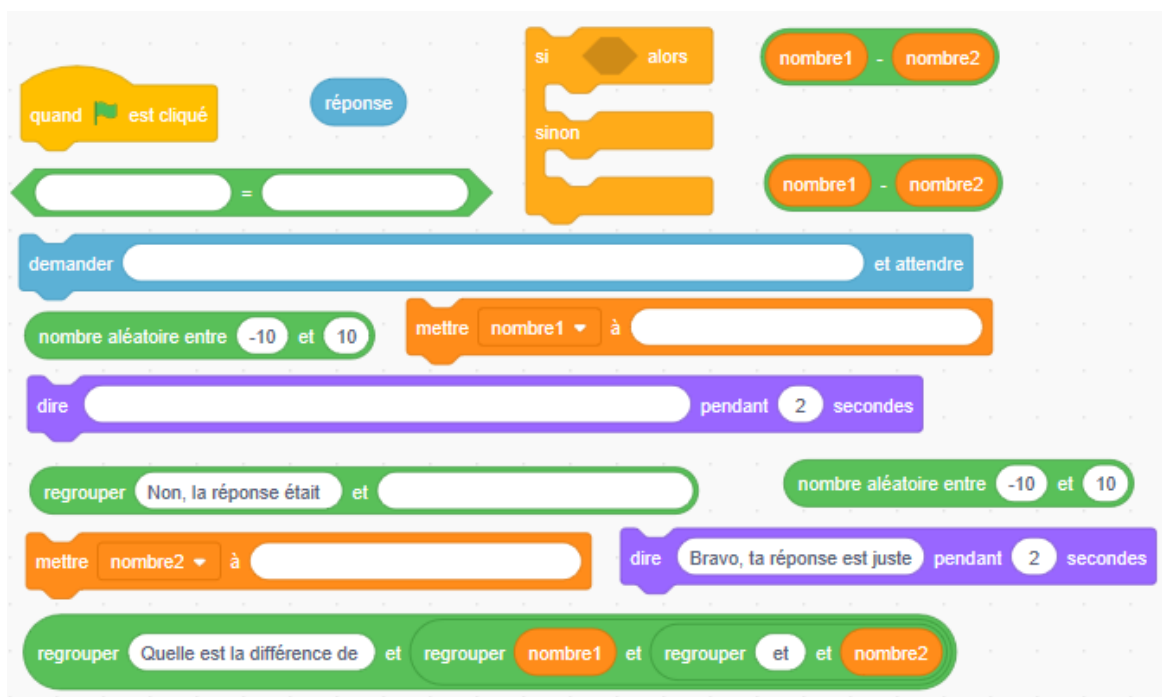
Partie n°2 : Réaliser un test / mettre en œuvre une instruction conditionnelle

Exercice n°1 : Un peu de calcul mental

Nolan souhaite s'entraîner au calcul mentale. Il décide de créer un algorithme qui, en fonction de deux nombres choisis au hasard entre -10 et 10 :

- lui demande de donner la différence,
- lui indique si sa réponse est juste ou fausse,
- indique la bonne réponse si le résultat donné est faux.

Voici dans le désordre les instructions que Nolan a utilisé sous Scratch ;



Remet les instructions dans le bon ordre puis traduit les en langage naturel

Script en Scratch	Langage naturel
Quand drapeau est cliqué Mettre nombre1 à nombre aléatoire entre -10 et 10 Mettre nombre2 à nombre aléatoire entre -10 et 10 Demander regrouper « Quelle est la différence de » et regrouper nombre1 et regrouper « et » et nombre2 et attendre. Si réponse = nombre1 – nombre2 alors Dire Bravo, ta réponse est juste pendant 2 Secondes Sinon Dire regrouper « Non, la réponse était » et nombre1 – nombre2 pendant 2 seconde.	Début algorithme Variables : nombre1, entier nombre2, entier réponse, entier Affecter un nombre aléatoire entre -10 et 10 à nombre1 Affecter un nombre aléatoire entre -10 et 10 à nombre2 Afficher « Quelle est la différence de », nombre1, « et », nombre2 Lire réponse Si réponse = nombre1 – nombre2 Afficher « Bravo, ta réponse est juste » Sinon Afficher « Non, la réponse était », nombre1- nombre2 Fin si Fin algorithme

Exercice n°2 : Utiliser un algorithme pour comparer deux tarifs

Mathis souhaite s'inscrire dans un cours de dessin. Le club lui propose les deux tarifs suivants :

- Tarif A : Une cotisation annuelle de 245 € et la séance coûte 5 €.
- Tarif B : Entrée libre à 15 €.

1. Quel sera le montant à payé par Mathis s'il va 2 fois par mois au cours de dessin durant l'année avec chacune des formules ?

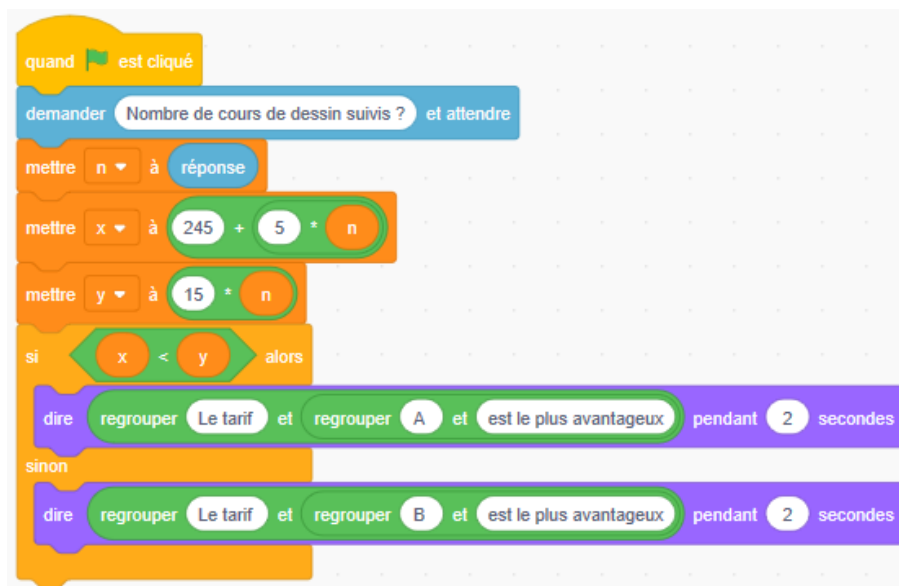
Tarif A : $245 + 5 \times 2 = 255$ €. Mathis devra payer 255 €.

Tarif B : $15 \times 2 = 30$ €. Mathis devra payer 30 €.

2. Quelle est la formule la plus avantageuse ?

La formule la plus avantageuse est ici la formule B car 30 € est nettement inférieur à 255 €.

Voici un programme incomplet qui, suivant le nombre n de séances suivies par Mathis durant l'année, affiche le tarif le plus avantageux.



3. Que représente dans ce programme les variables n , x et y ?

n : cette variable désigne le nombre de cours de dessins suivis durant l'année par Mathis

x : cette variable désigne le prix à payer avec la formule A par Mathis pour x cours de dessins suivis.

y : cette variable désigne le prix à payer avec la formule B par Mathis pour x cours de dessins suivis.

4. Compléter le programme.

5. Exécuter ce programme pour compléter le tableau suivant.

n	12	20	23	24	25	26	30	35
Tarif le plus avantageux	B	B	B	B	A	A	A	A

Exercice n°3 : Calculer un coût à partir d'un tarif évolutif

Dans une agence, le coût de la location journalière d'une voiture s'élève à 75 €. Ce tarif permet de parcourir 200 km. Chaque kilomètre supplémentaire entre 201 et 300 km coûte 0,28 €. Chaque kilomètre supplémentaire au-delà de 300 km coûte 0,45 €.

1. Calculer le coût d'une location pour 350 km parcourus.

350 km se décompose en :

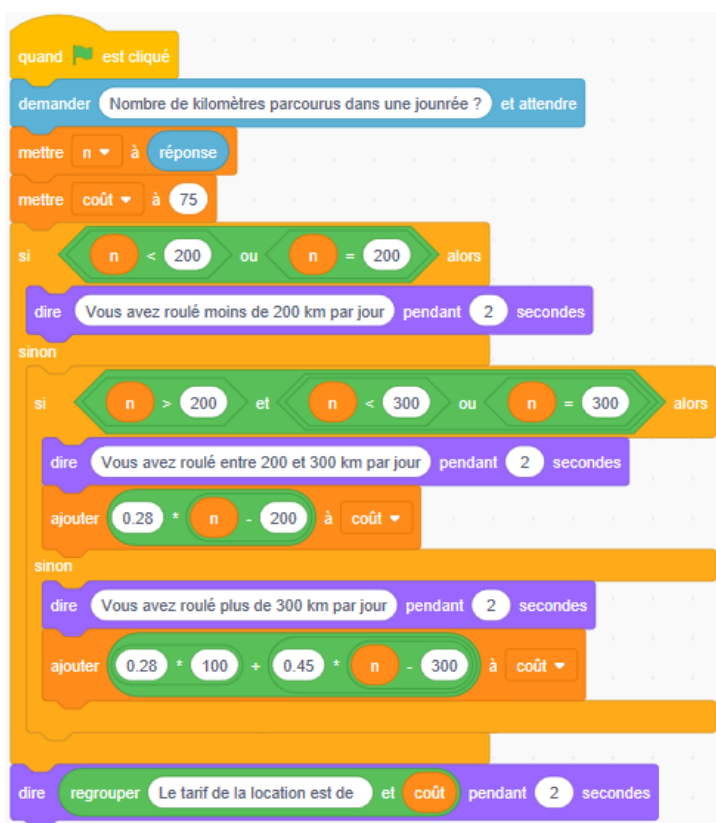
- 200 km compris dans le forfait,
- puis 100 km à 0,28 € le km,
- puis 50 km à 0,45 € le km.

Le coût d'une location est donc : $75 + 100 \times 0,28 + 50 \times 0,45 = 125,5$ €

Voici un programme incomplet qui, suivant le nombre x de kilomètres parcourus dans la journée, affiche le coût de la location.

2. Compléter le programme et le traduire en langage naturel

Script Scratch



Langage naturel

Début algorithme

Variables : n, entier
coût, entier

Afficher « Nombre de km parcourus dans une journée ? »
Lire n
coût prend la valeur 75
si $n \leq 200$
 afficher « Vous avez roulé moins de 200 km »
sinon
 si $n > 200$ et $n \leq 300$
 afficher « Vous avez roulé entre 200 et 300 km »
 coût = coût + $0,28(n - 200)$
 sinon
 afficher « Vous avez roulé plus de 300 km »
 coût = coût + $0,28 \times 100 + 0,45(n - 200)$
 Fin si
Fin si
Afficher « Le tarif de la location est de », coût
Fin algorithme

3. Exécuter ce programme pour compléter le tableau suivant.

Distance parcourue	199	200	201	299	300	301	340	400
Prix en €	75	75	75,28	102,72	103	103,45	121	148

Exercice n°4 : Synthèse sur les instructions conditionnelles

<p><u>Algorithme n°1</u></p> <p>Début algorithme</p> <p>Variable : réponse : texte</p> <p>Ecrire « Voulez-vous jouer ? oui ou non ? »</p> <p>Lire réponse</p> <p>Si réponse = 'oui'</p> <p> Ecrire « C'est parti ! »</p> <p>Sinon</p> <p> Ecrire « Dommage »</p> <p>Fin si</p> <p>Fin algorithme</p>	<p>Qu'obtient-on si l'utilisateur répond oui ? C'est parti !</p> <p>Qu'obtient-on si l'utilisateur répond non ? Dommage</p> <p>Qu'obtient-on si l'utilisateur répond bof ? Dommage</p> <p>Comment modifier l'algorithme pour qu'il écrive « Je n'ai pas compris » le cas échéant ?</p> <p>Sinon</p> <p> Si réponse = 'non'</p> <p> Ecrire « Dommage »</p> <p> Sinon</p> <p> Ecrire « Je n'ai pas compris »</p> <p> Fin si</p>
<p><u>Algorithme n°2</u></p> <p>Début algorithme</p> <p>Variable(s) :</p> <p>Ecrire « Entrer deux nombres ... »</p> <p>Ecrire « Quel est le premier ? »</p> <p>Lire nombre_1</p> <p>Ecrire « Quel est le deuxième ? »</p> <p>Lire nombre_2</p> <p>Si nombre_1 < nombre_2</p> <p> Ecrire « Le plus petit est », nombre_1</p> <p>Sinon</p> <p> Ecrire « Le plus petit est », nombre_2</p> <p>Fin si</p> <p>Fin algorithme</p>	<p>Quelles sont les variables de cet algorithme ? nombre_1 et nombre_2</p> <p>Qu'obtient-on si l'utilisateur répond 12 puis 5 ? Le plus petit est 5</p> <p>Qu'obtient-on si l'utilisateur répond 12 puis 12 ? Le plus petit est 12</p> <p>Comment modifier l'algorithme pour qu'il écrive « Les valeurs sont identiques » le cas échéant ?</p> <p>Sinon</p> <p> Si nombre_1 > nombre_2</p> <p> Ecrire « Le plus petit est », nombre_2</p> <p> Sinon</p> <p> Ecrire « Les valeurs sont identiques »</p> <p> Fin si</p> <p>Fin si</p>
<p>Lors d'un examen un candidat obtient le résultat suivant :</p> <ul style="list-style-type: none">- Reçu si sa moyenne est au moins de 10.- Admis à l'oral si sa moyenne est entre 8 et 10.- Recalé si sa moyenne est en dessous de 8. <p>L'examen comporte deux épreuves :</p> <ul style="list-style-type: none">- Epreuve écrite_1 : coefficient 2- Epreuve orale : coefficient 1 <p>Compléter l'algorithme pour que celui-ci indique la moyenne de l'élève ainsi que le bilan du conseil de classe.</p>	<p><u>Algorithme n°3</u></p> <p>Début algorithme</p> <p>Ecrire « Résultat de la première épreuve ? »</p> <p>Lire x</p> <p>Ecrire « Résultat de la deuxième épreuve ? »</p> <p>Lire y</p> <p>m prend la valeur $(x + y) / 2$</p> <p>Si $m \geq 10$</p> <p> Ecrire « Reçu ! »</p> <p>Sinon</p> <p> Si $m \geq 8$</p> <p> Ecrire « Admis à l'oral »</p> <p> Sinon</p> <p> Ecrire « Recalé ! »</p> <p> Fin si</p> <p>Fin si</p> <p>Fin algorithme</p>

Partie n°3 : Boucle bornée et non bornée

Exercice n°1 : Faire des économies

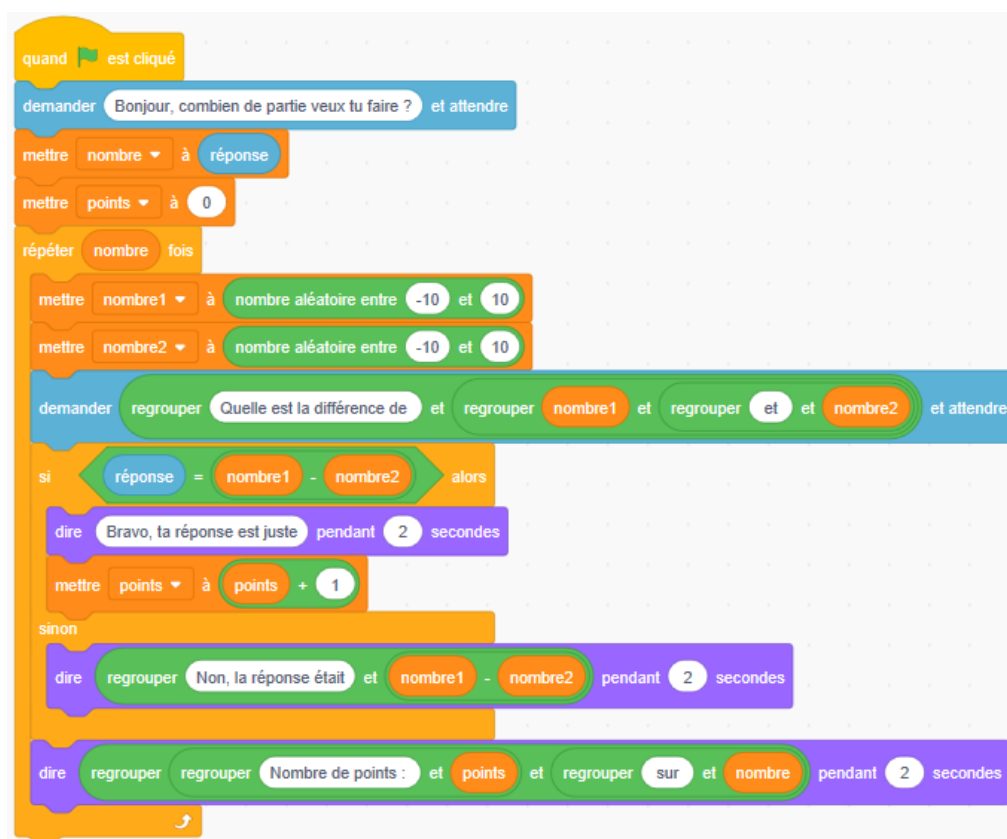
Julien reçoit chaque semaine 15€ d'argent de poche. Il en dépense les deux tiers et économise le reste. Actuellement il a 23€ dans sa tirelire.

Le but de cet exercice est d'établir un script qui permette de déterminer le nombre de semaines à attendre avant que Julien puisse s'acheter une console de jeux vidéo coûtant 200 €. Voici les instructions à utiliser :



1. Quelles sont les variables de ce programme ?
n et tirelire
2. Ecrire le script Scratch correspondant au problème précédent puis le traduire en langage naturel.

Instruction Scratch	Langage naturel
	Début algorithme <i>On note aussi « Fin algorithme » lorsque celui-ci doit s'arrêter.</i>
	Afficher le message : « » Lire Réponse
	Affecter « » à 'a' (ou 'a' prend la valeur ...)
	Afficher « phrase »
	Tant que <i>condition non réalisée</i> <i>Instruction</i> Fin tant que



Exercice n°2 : Table de multiplication

On souhaite écrire un script permettant d'afficher la table de 7 de 7×1 à 7×10 . Pour pouvoir laisser la trace écrite de la table de multiplication, il est nécessaire de créer dans Scratch une « liste » qui contiendra les différents résultats de la table de 7. Pour ce faire, il suffit d'aller dans la catégorie « variable » puis de sélectionner « créer une liste ». Voici la liste des blocs utilisés dans Scratch.



1. Ecrire le script Scratch correspondant au problème précédent puis le traduire en langage naturel.



2. Modifier le programme précédent afin qu'il affiche uniquement la partie de la table de 7 choisie par l'utilisateur.



Exercice n°3 : Synthèse sur les boucles

1. Pour les algorithmes 1 et 2, indiquer le résultat de chaque instruction.

Algorithme n°1

Début algorithme

Variables r : texte

Ecrire « Veux-tu jouer ? oui ou non ? »

Lire r

Tant que r != 'oui' ou r != 'non'

 Ecrire « Je n'ai pas compris »

 Lire réponse

Fin tant que

Fin algorithme

.....

Création de la variable r de type texte

Affichage du texte « Veux-tu jouer ? oui ou non ? »

On récupère la réponse de l'utilisateur

Test du contenu de la variable réponse

Si réponse ≠ oui ou non on écrit « Je n'ai pas compris.

On récupère à nouveau la réponse de l'utilisateur

Puis on recommence le test ...

Lorsque la variable r contient oui ou non, on sort de la boucle tant que

.....

Algorithme n°2

Variables n : numérique

Début algorithme

 n prend la valeur 0

 Ecrire « Entrer un nombre entre 10 et 20 »

 Tant que n < 10 ou n > 20

 Lire n

 Fin tant que

Fin algorithme

Création de la variable n de type numérique

.....

n = 0

Affichage du texte « Entrer un nombre ... »

Test du contenu de la variable n

On récupère la réponse de l'utilisateur

On sort de la boucle lorsque n est entre 10 et 20

.....

2. Modifier l'algorithme n°2 afin qu'il écrive :

- « on a dit plus grand que 10 ! » si le nombre entré est inférieur à 10
- « on a dit plus petit que 20 ! » si le nombre entré est supérieur à 20

Algorithme n°3

Variables n : numérique

Début algorithme

 n prend la valeur 0

 Ecrire « Entrer un nombre entre 10 et 20 »

 Tant que n < 10 ou n > 20

 Lire n

 Si n < 10

 Ecrire « on a dit plus grand que 10 ! »

 Fin si

 Si n > 20

 Ecrire « on a dit plus petit que 20 ! »

 Fin si

 Fin tant que

Fin algorithme