

0
1
0
0
1
1
0
1
0
1
0
1
0
1
1

</>

{ _ _ }

< _

</>

{

0 1
1 0
0 1
1 0
1 1
0 0
0 1
0 1
1 0
1 1
1 0
1 1

{



PROGRAMMATION I

Session 8.4





- Variables et Opérateurs
- Turtle + for/while
- Boucles + tableaux
- Fonction + Turtle
- Fonction + return

Voici 5 tâches de programmation qui permettront de déterminer si tu as validé le Niveau 1.

Pour réussir, il te faudra compléter au moins 3 tâches sur 5, dont la tâche 4 ou 5.



1 – Rends toi sur l'URL Classroom : **cutt.ly/tumo-lyon-lab-1**

2 – Connecte-toi à ton compte GitHub

Sign in to GitHub
to continue to GitHub Classroom

Username or email address

Password [Forgot password?](#)

Sign in



3 – Une fois connecté, sélectionne ton identifiant de Tumoien

4 – Rejoins le Lab en cliquant sur « Accept this assignment »

Join the classroom:

Programming Lab - Level 1

To join the GitHub Classroom for this course, please select yourself from the list below to associate your GitHub account with your school's identifier (i.e., your name, ID, or email).

3

Can't find your name? [Skip to the next step](#) →

Identifiers

anthony	>
carol	>
jean-sebastien	>
jimmy	>
jérémy	>
tuba	>

Programming Lab - Level 1

Accept the assignment —

Lab Niveau 1

Once you accept this assignment, you will be granted access to the `lab-niveau-1-jscolas` repository in the `tumo-arao` organization on GitHub.

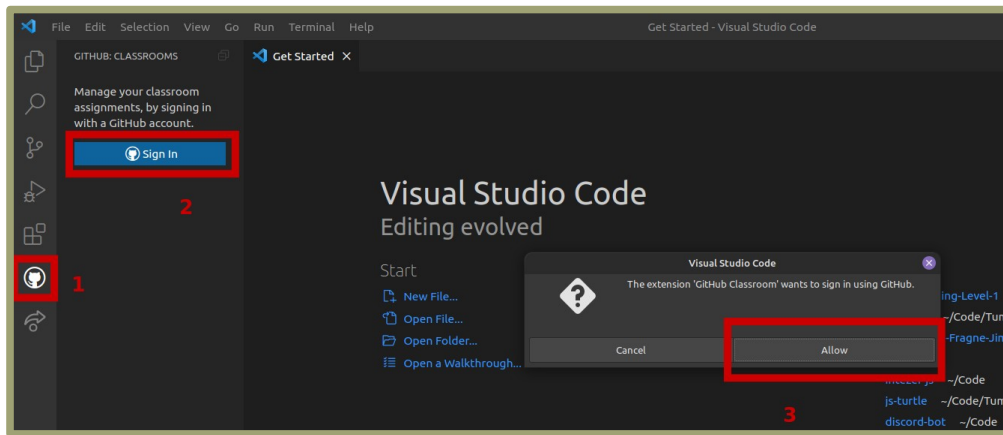
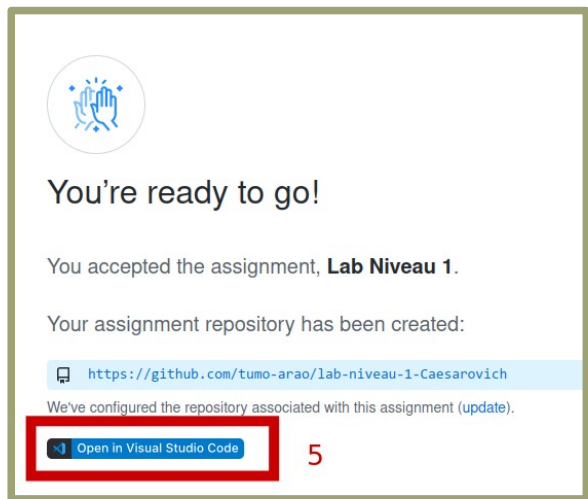
4

Accept this assignment



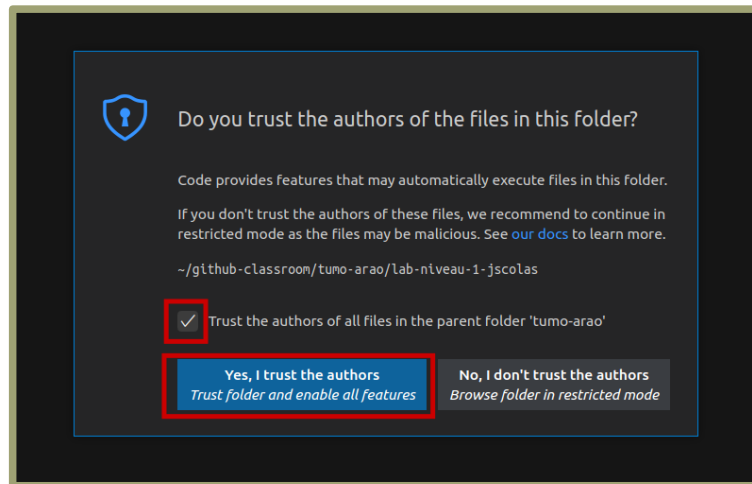
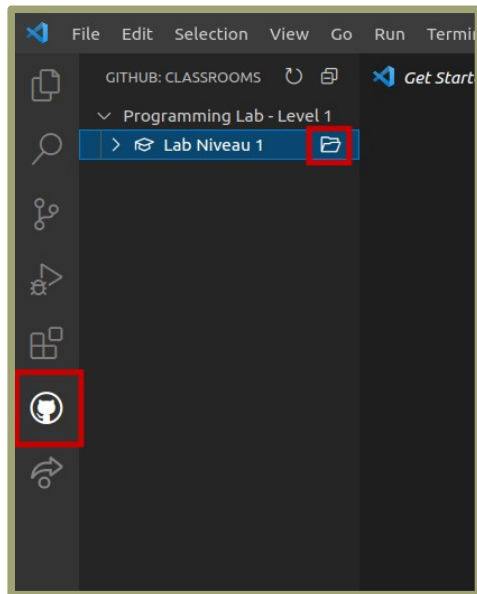
5 – Une fois ton Repo créé, clique sur « **Open in VSCode** »

Quand VSCode est ouvert, va dans l'onglet « **1. GitHub** », appuies sur « **2. Sign In** » puis « **3. Allow** ».
Dans le navigateur acceptes toutes les demandes.





Une fois GitHub Classrooms est connecté, clique sur l'icone correspondant à ce lab.



Clique sur « Oui je fais confiance aux auteurs »

C'est parti !



Écris un programme qui calcule **y** et l'affiche dans la console.
Les variables **m**, **k** et **n** doivent être saisies par l'utilisateur
au moment de l'exécution du programme.

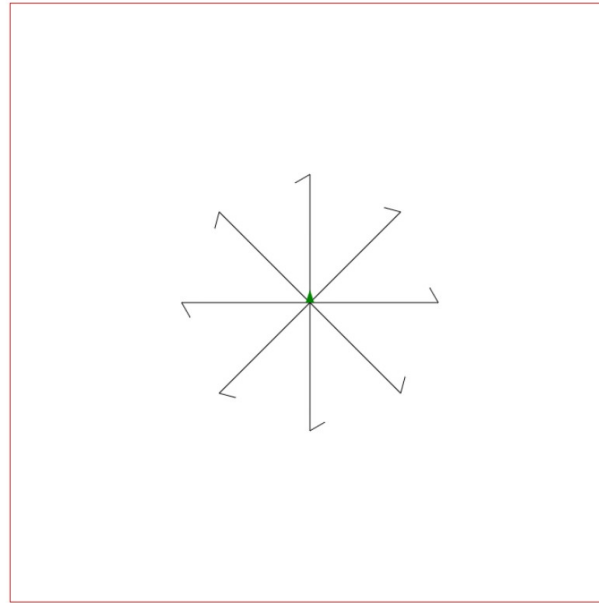
Le programme doit fonctionner pour **m**, **k** et **n** fractionnaires
(Nombres à virgule). Ton programme doit empêcher les
divisions par 0.

$$y = m^5 - 100/k + n * 5$$



Dessine l'image suivante utilisant des boucles (for ou while)

Canvas





Crées un programme qui remplis un tableau avec tous les nombres impairs dans l'intervalle de 4 à 671.

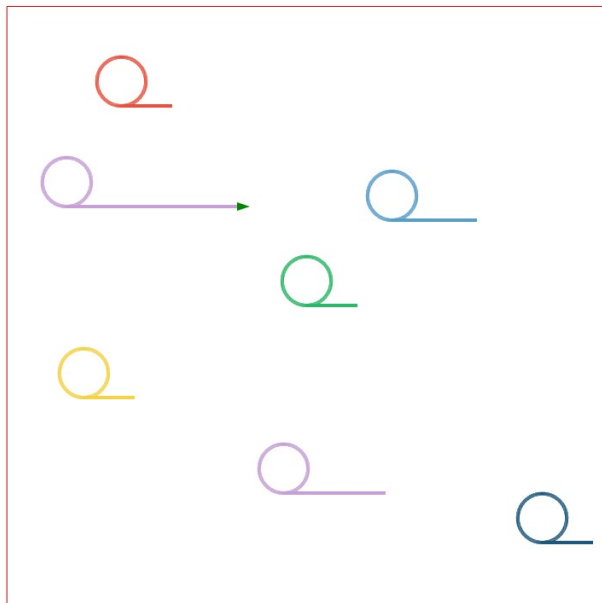
Le programme doit ensuite afficher le tableau dans la console.

Tâche 4



Écris un programme qui dessine l'image suivante. La création et l'utilisation d'une fonction sont obligatoires. Trouve l'information sur les couleurs ou la position dans le tableau ci-dessous :

Canvas



X, Y	Longueur	(RVB)
100, 100	100	84, 153, 199
0, 0	60	40, 180, 99
-217, 234	60	231, 76, 60
276, -278	60	26, 82, 118
-261, -108	60	244, 208, 63
-27, -220	120	195, 155, 211
-281, 116	200	195, 155, 211



Écris une fonction qui prend pour argument un nombre et détermine si le nombre saisi est multiplié par 4 ou non.

L'utilisateur saisira le nombre au moment de l'exécution du programme. Le nombre sera donc ensuite passé en **argument** à ta fonction.

Si le nombre est un multiple de 4, **retourne** "true", sinon **retourne** "false".

Utilise la fonction `isNaN()` pour vérifier si la valeur saisi est bien un nombre. Si la valeur n'est pas un nombre, la fonction devra **retourner** « Ce n'est pas un nombre ».

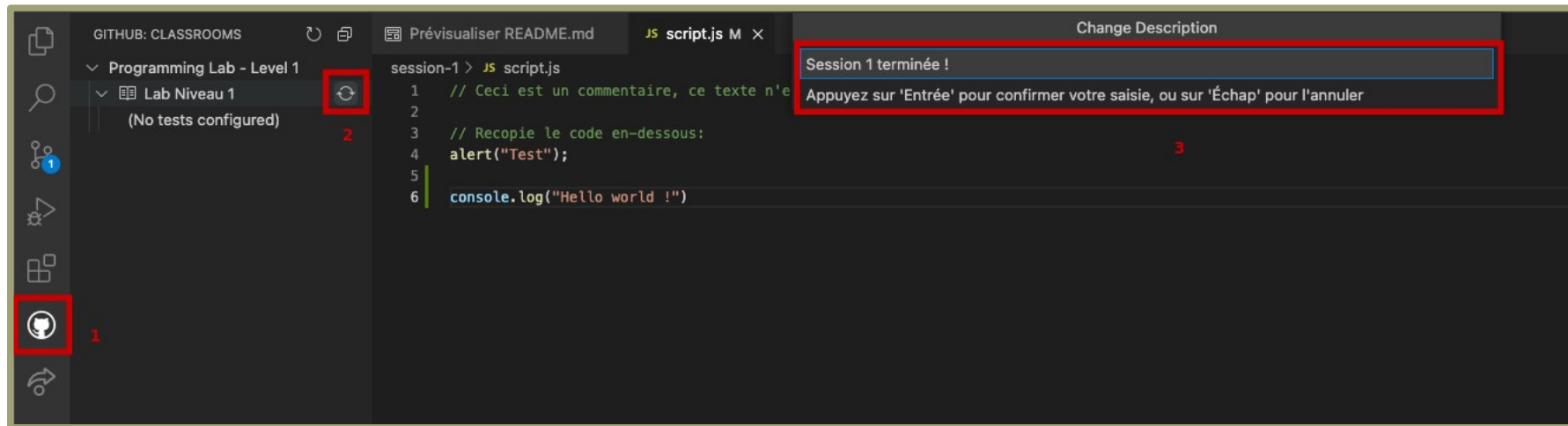
Teste la fonction pour les valeurs suivantes :

- 100
- 9
- abc

Enregistrer ton travail

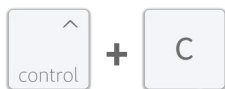


Retourne dans l'onglet GitHub et clique sur l'icone correspondant à ce lab.



Ressources supplémentaires

Raccourcis utiles



+

C

COPIER



+

X

COUPER



+

V

COLLER



+

S

ENREGISTRER



+

Z

REVENIR EN ARRIERE



+

Y

REVENIR EN AVANT



+

F

RECHERCHER



+

H

REEMPLACER



{ / }



[/]



Mac



=>

