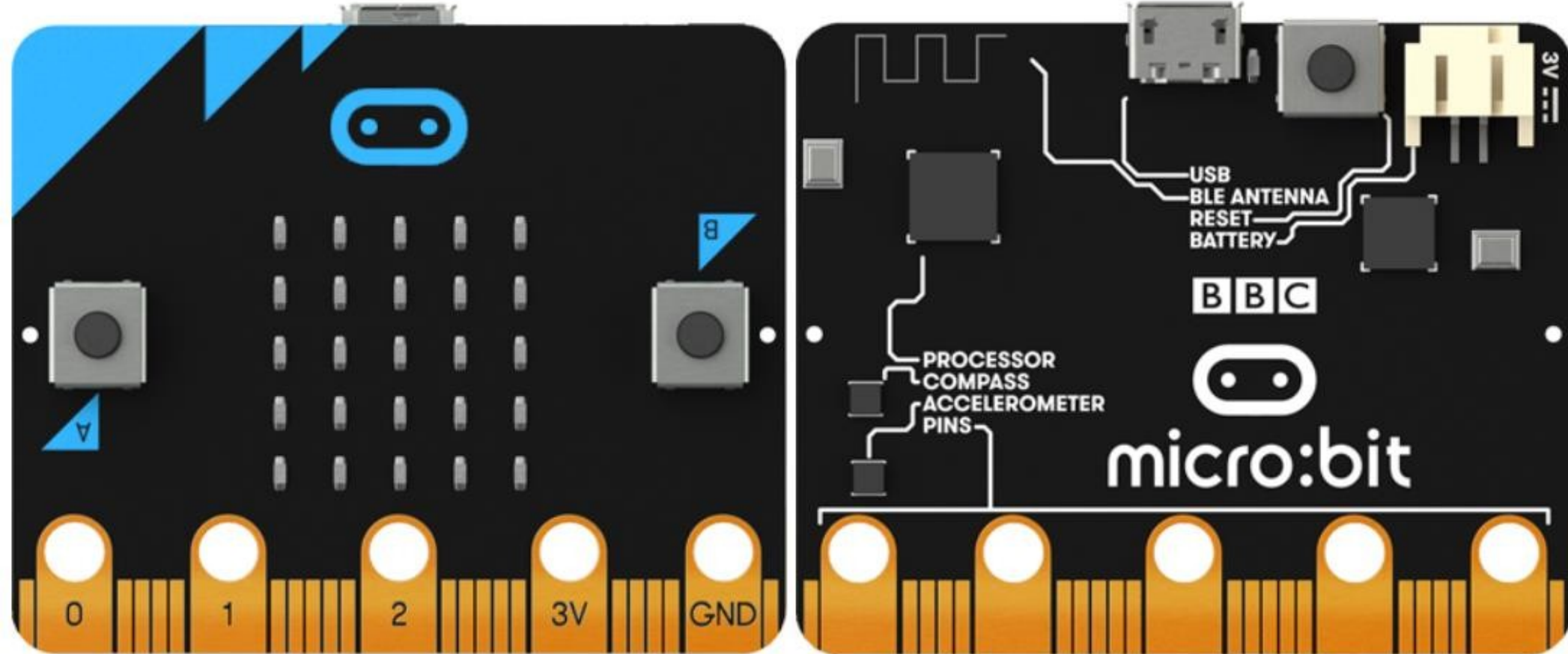


Présentation carte Micro:bit



La carte contient de nombreux capteurs, voici ceux que nous utiliserons dans ce document :

- Deux boutons programmables, désignés « A » et « B » sur la carte
- afficheur digital carré de 25 LED (5 × 5) rouges programmables pouvant servir d'affichage, notamment pour des motifs animés, du texte alphanumérique déroulant

Micro :bit découverte

Nous allons découvrir la carte Micro:bit
avec le site Vittascience en ligne

<https://fr.vittascience.com>

Ce site permet de travailler avec des
blocs ou en python sur Microbit et il
dispose d'un simulateur intégré

Ouvrez le Site



Vittascience Micro:bit



The screenshot shows the Vittascience website interface. The browser address bar displays <https://fr.vittascience.com>. The navigation bar includes the Vittascience logo, 'Boutique', 'Communauté', 'Programmer', 'Ressources', 'Météo', and 'À propos'. The 'Programmer' dropdown menu is open, showing 'Arduino', 'Microbit', and 'Python'. The main banner features a background image of a hot air balloon over a forest, with the text 'SCIENCE ET PARTAGE' and the tagline 'Pour une approche collaborative et humaine de la Science.' Below this is an orange button labeled 'Carte des projets'. At the bottom, a cookie consent banner is visible with the text 'En poursuivant votre navigation, vous acceptez le dépôt de cookies tiers destinés à vous proposer une meilleure expérience utilisateur. Plus d'informations' and an 'Accepter' button.

https://fr.vittascience.com/microbit/

Cliquez sur Microbit dans Programmer

Vittascience Micro:bit

Nous allons découvrir la carte en utilisant les ressources internes de base de la carte.

En entrée les 2 boutons-poussoirs BP_A et BP_B

En sortie les 25 LED rouges (5 lignes, 5 colonnes)

Vous pourrez **simuler la carte Microbit avec la partie Droite**

A gauche , la partie programmation avec 3 zones :

À droite
Les blocs
disponibles

Au Milieu
Le programme
sous forme de
blocs

À gauche
Le programme
en python

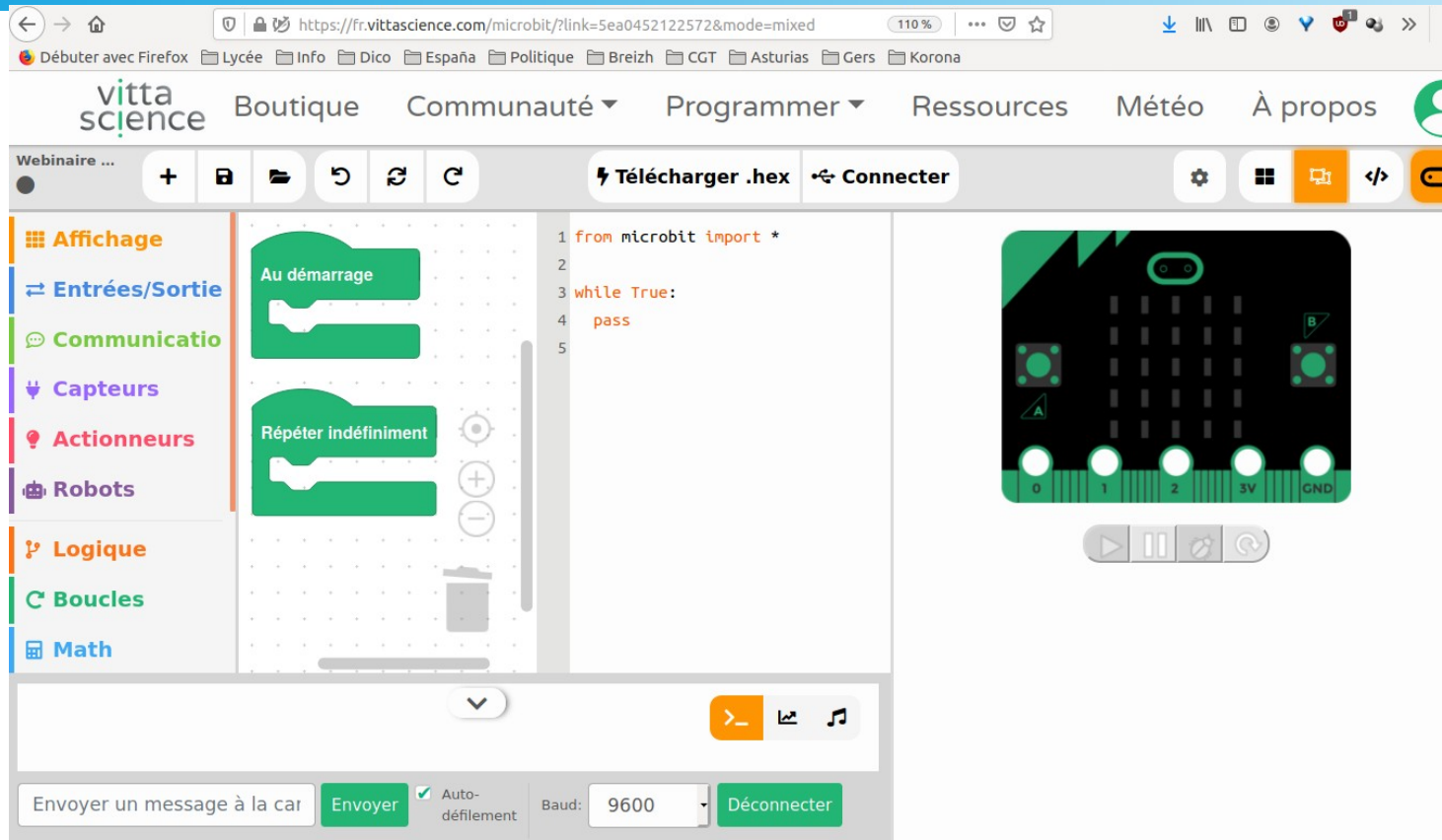
The screenshot shows the Vittascience Micro:bit web interface. The top navigation bar includes links for 'Boutique', 'Communauté', 'Programmer', 'Ressources', 'Météo', and 'À propos'. The main interface is divided into three sections:

- Left Panel (Blocks):** A vertical list of block categories: 'Affichage', 'Entrées/Sortie', 'Communication', 'Capteurs', 'Actionneurs', 'Robots', 'Logique', 'Boucles', and 'Math'.
- Middle Panel (Block-based Programming):** A workspace for building programs using blocks. It contains two green blocks: 'Au démarrage' (At startup) and 'Répéter indéfiniment' (Repeat indefinitely).
- Right Panel (Python Code Editor):** A text editor for writing Python code. The code shown is:

```
1 from microbit import *
2
3 while True:
4     pass
5
```

At the bottom, there is a status bar with a text input 'Envoyer un message à la car', an 'Envoyer' button, a checkbox for 'Auto-défilement', a 'Baud' rate selector set to '9600', and a 'Déconnecter' button.

Vittascience Micro:bit

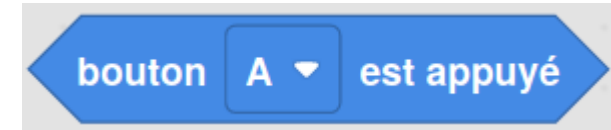
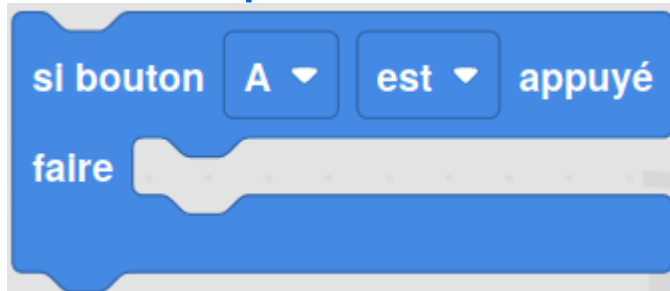


Le bloc « démarrage » n'est fait qu'une seule fois au début, il sert à initialiser des variables ou des fonctions.

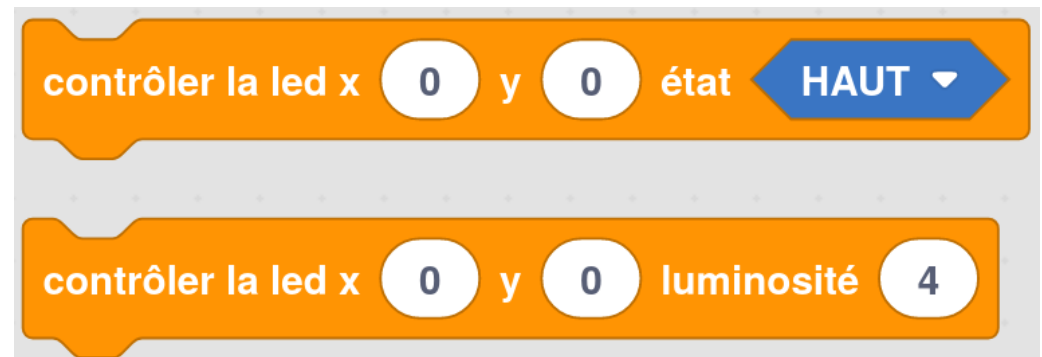
Le bloc « répéter indéfiniment » est une boucle infinie dans lequel on placera le programme principal, il correspond en python à la boucle sans fin « while True »

Vittascience Micro:bit Exo1

- Exercice n°1 Allumer une LED si le bouton A est appuyé
- Si on clique sur Entrées , 2 correspondent au bouton A



- Si on clique sur LED, nous utiliserons ces 2 cas par la suite



Pour information la LED $x = 0, y = 0$ correspond à la LED en haut à gauche.

Vittascience Micro:bit Exo1

The screenshot shows the Vittascience Micro:bit programming environment. The top navigation bar includes links for 'Boutique', 'Communauté', 'Programmer', 'Ressources', 'Météo', and 'À propos'. The main workspace is divided into three sections: a block-based editor on the left, a Python code editor in the center, and a virtual Micro:bit board on the right.

Block-based editor: The program starts with 'Au démarrage' (At startup), followed by a 'Répéter indéfiniment' (Repeat indefinitely) loop. Inside the loop, there is a 'si bouton A est appuyé' (if button A is pressed) condition. If true, it executes 'contrôler la led x 0 y 0 état HAUT' (control the LED at x=0, y=0, state ON).

Python code editor: The code is as follows:

```
1 from microbit import *
2
3 while True:
4     if button_a.is_pressed():
5         display.set_pixel(0,0,9)
6
```

Virtual Micro:bit board: The board is shown with its 5x5 LED matrix. The bottom row of pins is labeled 0, 1, 2, 3V, and GND. The board is currently displaying a green pixel at the top-left corner (0,0).

Simulez ce montage très simple en cliquant sur la carte à droite sur le Bouton A.

1) Que constatez-vous ?

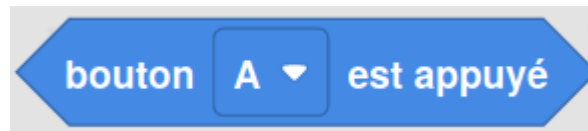
2) Comment peut-on éteindre la LED ?

Vittascience Micro:bit Exo1

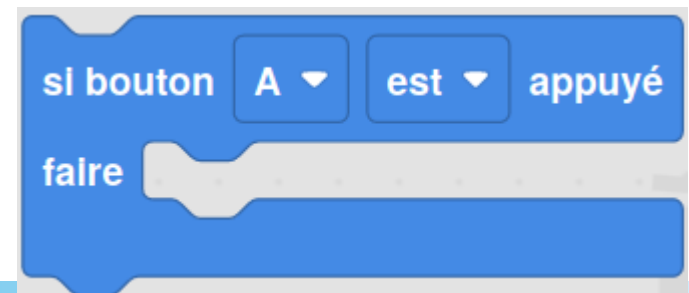
La carte n'est pas intelligente, si vous ne lui dites pas quand éteindre la LED, elle ne le fera pas.

Exercice n°1 :

- Allumer une LED si le bouton A est appuyé
- Devient **si** le bouton A est appuyé **alors** Allumer la LED(0,0) **sinon** éteindre la LED(0,0)
- Nous utiliserons dans la Logique le deuxième bloc avec si, sinon
- **Remarque** pour le bouton A nous devons utiliser ce bloc



nous ne pouvons plus utiliser ce bloc qui ne possède pas de sinon



Vittascience Micro:bit Exo1

The screenshot displays the Vittascience Micro:bit programming environment. At the top, the navigation bar includes 'vitta science', 'Boutique', 'Communauté', 'Programmer', 'Ressources', 'Météo', and 'À propos'. Below this, a toolbar contains icons for file management and execution, along with buttons for 'Télécharger .hex' and 'Connecter'. The left sidebar lists various function categories: Affichage, Entrées/Sorties, Communication, Capteurs, Actionneurs, Robots, Logique, and Boucles. The main workspace is divided into three sections: 1. Block-based programming: A script starting with 'Au démarrage' followed by an infinite loop 'Répéter Indéfiniment'. Inside the loop is an 'if' block checking 'bouton A est appuyé'. If true, it sets the LED at (0,0) to 'HAUT'; otherwise, it sets it to 'BAS'. 2. Python code: The corresponding code is shown on the right:

```
1 from microbit import *
2
3 while True:
4     if button_a.is_pressed():
5         display.set_pixel(0,0,9)
6     else:
7         display.set_pixel(0,0,0)
8
```

 3. Simulation: A visual representation of the Micro:bit board on the right, showing the LED matrix with a green pixel at (0,0) and a green bar at the bottom. Below the board are playback controls (play, pause, stop, reset).

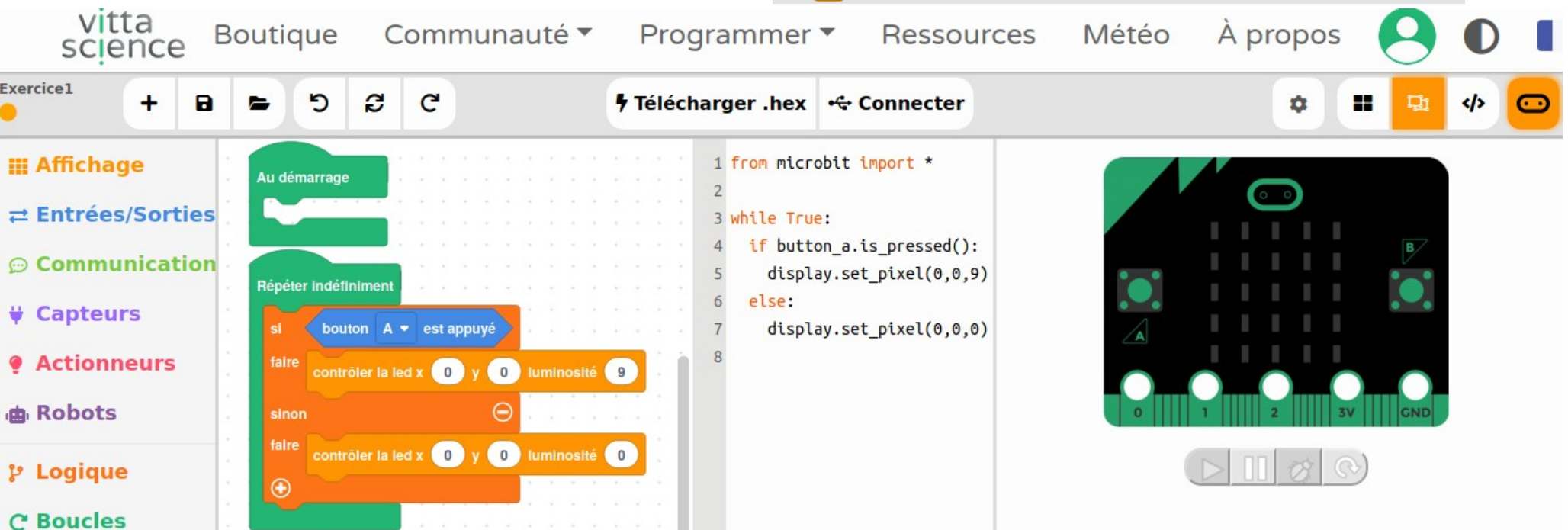
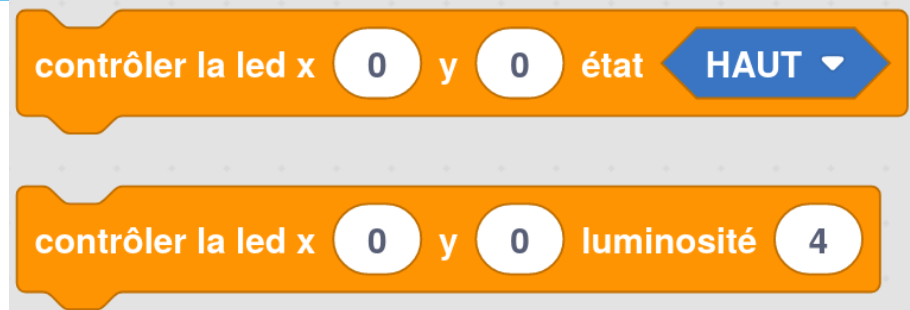
Réalisez ce programme et simulez le avec la carte à droite
Conclure.

Regardez le programme python et faites le lien entre les
blocs et le code.

Vittascience Micro:bit Exo1 bis

On peut remplacer ce bloc

Par celui-ci



Comparez le code python généré et celui de la page précédente, que constatez-vous ?

Simulez ce montage puis modifiez la luminosité 9 par 5 que constatez-vous ?

Vittascience Micro:bit Exo2

- Nous allons travailler dans cet exercice2 avec les 2 boutons poussoirs pour réaliser une commande Marche/Arrêt comme on en trouve dans beaucoup de système industriel.
- Si j'appuie sur le bouton Vert (*1 Marche*) puis je relâche :
Le système se met en marche.
- Si j'appuie sur le bouton Rouge (*0 Arrêt*) puis je relâche :
Le système s'arrête.
- Carte Micro:bit :
- Bouton_A remplace bouton Rouge (Arrêt)
Bouton_B remplace bouton Vert (Marche)
La LED(0,0) remplace le système



Vittascience Micro:bit Exo2

- Assemblez les blocs pour réaliser l'exercice 2 et simuler le !

The screenshot shows the Vittascience Micro:bit programming environment. The top navigation bar includes links for 'vitta science', 'Boutique', 'Communauté', 'Programmer', 'Ressources', 'Météo', and 'À propos'. The main workspace is divided into three sections: a left sidebar with category icons (Affichage, Entrées/Sorties, Communication, Capteurs, Actionneurs, Robots, Logique, Boucles, Math), a central block-based editor, and a right section for Python code and a simulation.

Block-based code:

- Au démarrage** (When started) block.
- Répéter indéfiniment** (Repeat indefinitely) loop containing:
 - si bouton A est appuyé** (if button A is pressed) block, followed by a **faire** (do) block.
 - si bouton B est appuyé** (if button B is pressed) block, followed by a **faire** (do) block.
- contrôler la led x 0 y 0 état BAS** (control LED x=0, y=0, state LOW) block.
- contrôler la led x 0 y 0 état HAUT** (control LED x=0, y=0, state HIGH) block.

Python code:

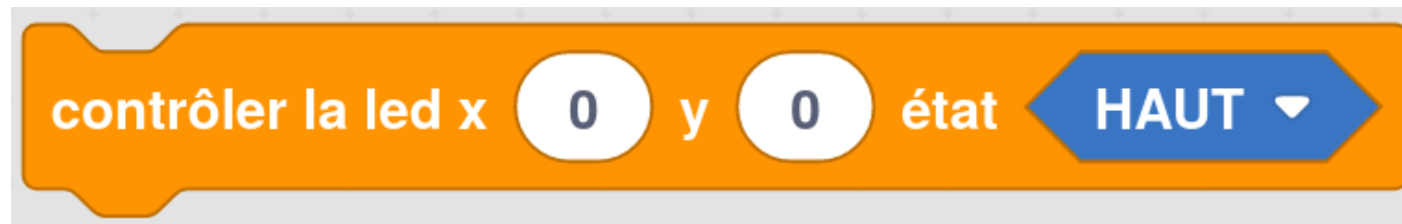
```
1 from microbit import *  
2  
3 while True:  
4     pass  
5
```

Simulation: A visual representation of the Micro:bit board with the LED matrix and pins (0, 1, 2, 3V, GND) shown.

Profitez de ces exercices simples pour faire le lien entre bloc et code python

Vittascience Micro:bit Exo2


- **Changez la LED**, on vous demande d'allumer et d'éteindre la LED au centre de la carte.
- Combien la carte Micro:bit a-t-elle de LED ?
- Combien en horizontal ? (x)
- Combien en vertical ? (y)
- Cliquez sur le « 0 » de x pour le modifier,












- Modifier aussi la valeur de y
- Simulez votre solution.

Vittascience Micro:bit Exo3

- Télérupteur : A chaque appui sur le BP, la LED change d'état.
- Réaliser le montage et simuler le, conclure.

vittascience Boutique Communauté ▾ Programmer ▾ Ressources Météo À propos 

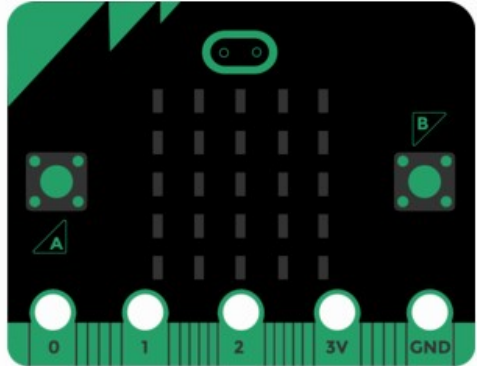
Exercice1       Télécharger .hex Connecter   

Affichage
Entrées/Sorties
Communication
Capteurs
Actionneurs
Robots
Logique

Au démarrage
fixer bascule à 0

Répéter indéfiniment
si bouton A est appuyé
faire
fixer bascule à 9 - bascule
contrôler la led x 0 y 0 luminosité bascule
pause 100 milliseconde(s)

```
1 from microbit import *  
2  
3 bascule = 0  
4  
5 while True:  
6     if button_a.is_pressed():  
7         bascule = 9 - bascule  
8         display.set_pixel(0,0,bascule)  
9         sleep(100)  
10
```



Faites des appuis brefs sur BP_A.

- Que se passe-t-il si l'appui sur le BP_A est long ?
- A quoi sert la variable bascule ?
- Quelles sont les valeurs qu'elle peut prendre ?
- Justifier ces valeurs ?

Vittascience Micro:bit Exo4

- Minuterie : A chaque appui sur le BP, La LED s'allume 3s
- En vous aidant des exercices 2 et 3, proposez une solution et simulez la.



Vittascience Micro:bit Exo5

- Guirlande de LED, réalisez le montage et simulez le. Précisez les positions x,y des LED qui s'allument.
- Expliquez la boucle For

The screenshot displays the Vittascience Micro:bit programming environment. The top navigation bar includes links for 'vitta science', 'Boutique', 'Communauté', 'Programmer', 'Ressources', 'Météo', and 'À propos'. The main workspace is divided into three sections: a left sidebar with category icons (Affichage, Entrées/Sorties, Communication, Capteurs, Actionneurs, Robots, Logique), a central block-based editor, and a right-hand Python code editor.

Block-based Program:

- Au démarrage** (When started):
- Répéter indéfiniment** (Repeat indefinitely):
 - compter avec** i de 0 à 4 par pas de 1
 - faire** (do):
 - contrôler la led x** i y 0 état HAUT
 - pause** 200 milliseconde(s)
 - contrôler la led x** i y 0 état BAS

Python Code:

```
1 from microbit import *
2
3 while True:
4     for i in range(5):
5         display.set_pixel(i,0,9)
6         sleep(200)
7         display.set_pixel(i,0,0)
8
```

Simulation: The right panel shows a simulated Micro:bit display with a 5x5 grid of LEDs. The top row (y=0) is lit with red LEDs at positions x=0, 1, 2, 3, and 4. The bottom of the panel shows a pin header with pins labeled 0, 1, 2, 3V, and GND.

Vittascience Micro:bit Exo6

- Guirlande de LED version 2, réalisez et simulez le montage Précisez les positions x,y des LED qui s'allument.
- Expliquez précisément la boucle For

The screenshot displays the Vittascience Micro:bit programming environment. The top navigation bar includes links for 'vitta science', 'Boutique', 'Communauté', 'Programmer', 'Ressources', 'Météo', and 'À propos'. The main workspace is divided into three sections: a left sidebar with category icons (Affichage, Entrées/Sorties, Communication, Capteurs, Actionneurs, Robots, Logique), a central area with a block-based program and a Python code editor, and a right sidebar showing a simulated Micro:bit board.

Block-based Program:

- Au démarrage** (When started)
- Répéter indéfiniment** (Repeat indefinitely)
- compter avec** (Count with): `i` de `4` à `0` par pas de `1`
- faire** (do):
 - contrôler la led x** `i` `y` `0` état **HAUT**
 - pause** `200` milliseconde(s)
 - contrôler la led x** `i` `y` `0` état **BAS**

Python Code:

```
1 from microbit import *
2
3 while True:
4     for i in range(4, -1, -1):
5         display.set_pixel(i,0,9)
6         sleep(200)
7         display.set_pixel(i,0,0)
8
```

Simulation: The right sidebar shows a simulated Micro:bit board with a 5x5 LED grid. The board is labeled with pins: 0, 1, 2, 3V, and GND. The LED grid shows a red light at position (4, 0) and a green light at position (0, 0).

Vittascience Micro:bit Exo7

- A partir des deux exercices précédents, créez une variable appelée j pour gérer les lignes, i sert pour les colonnes puis utilisez i et j pour allumer les 25 LED, les unes après les autres.
- Proposez un programme et simuler le.
- Variante, faire un serpent in avec les LED
- Pour faire le serpent in, pensez à vous servir de cette fonction mathématique
- On peut remplacer 2 par une variable



Vittascience Micro:bit Exo8

Remarquez comment sont réalisées les lettres avec les LED , changez la lettre N par d'autres lettres.

Regardez le code python, que signifie scroll ?

The screenshot shows the Vittascience Micro:bit programming environment. The top navigation bar includes links for 'vittascience', 'Boutique', 'Communauté', 'Programmer', 'Ressources', 'Météo', and 'À propos'. Below this is a toolbar with icons for adding blocks, saving, undo, redo, and a 'Télécharger .hex' button. The left sidebar contains a category menu with 'Affichage', 'Entrées/Sortie', 'Communication', 'Capteurs', 'Actionneurs', 'Robots', and 'Logique'. The main workspace is divided into three sections: a block-based code editor on the left, a Python code editor in the center, and a virtual Micro:bit display on the right. The block editor shows a 'Répéter indéfiniment' loop containing an 'afficher le texte' block with the letter 'N'. The Python editor shows the following code:

```
1 from microbit import *
2
3 while True:
4     display.scroll(str('N'))
5
```

The virtual Micro:bit display on the right shows the letter 'N' formed by red LEDs on a black background. Below the display are buttons for play, pause, reset, and a refresh icon.

Vittascience Micro:bit Exo9

Ici nous avons fait un N avec une image, on coche les zones à afficher. Regardez bien le code python

The screenshot shows the Vittascience Micro:bit programming environment. The top navigation bar includes links for 'vitta science', 'Boutique', 'Communauté', 'Programmer', 'Ressources', 'Météo', and 'À propos'. The main workspace is divided into three sections:

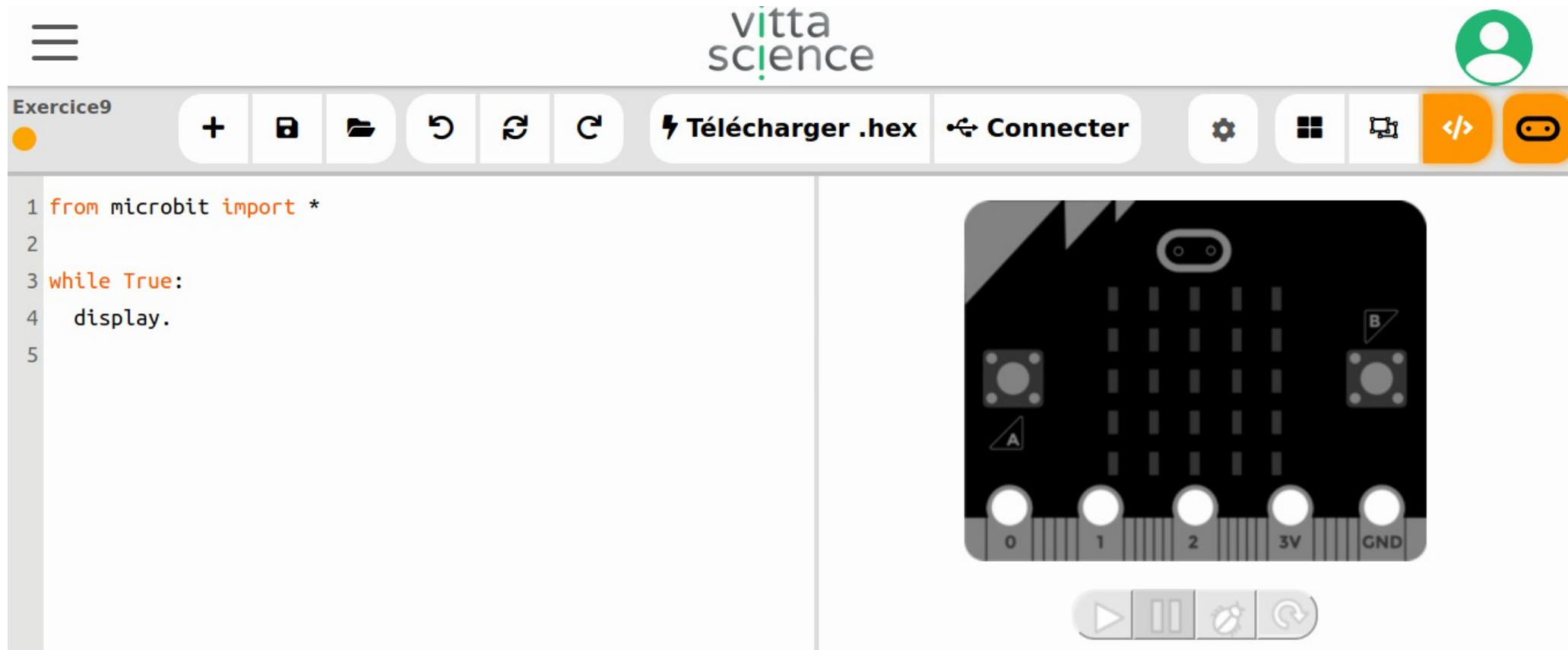
- Left Panel (Blocks):** A sidebar with categories like 'Affichage', 'Entrées/Sorties', 'Communication', 'Capteurs', 'Actionneurs', 'Robots', 'Logique', 'Boucles', 'Math', 'Texte', 'Variables', 'Listes', and 'Fonctions'. The 'Affichage' category is selected, showing a 'Répéter indéfiniment' (Repeat indefinitely) block containing an 'afficher l'image' (show image) block. The 'afficher l'image' block has a 5x5 grid of checkboxes, with the top row and the first column checked to form the shape of the letter 'N'.
- Center Panel (Code):** A Python script is displayed:

```
1 from microbit import *  
2  
3 while True:  
4     led_image = Image('90009:99009:90909:90099:90009')  
5     display.show(led_image)  
6
```
- Right Panel (Visual):** A visual representation of the Micro:bit LED display. The display shows the letter 'N' formed by red LEDs on a black background. The pins at the bottom are labeled 0, 1, 2, 3V, and GND. Below the display are control buttons for play, pause, reset, and refresh.

Vittascience Micro:bit Exo10

Pour finir, nous allons travailler directement en python avec le mode code.

A partir des 2 exemples précédents, proposer un code qui affiche la lettre 'N' (comme exo8) mais fixe (comme exo9)



The screenshot shows the Vittascience Micro:bit IDE interface. At the top, there is a header with the Vittascience logo and a user profile icon. Below the header is a toolbar with various icons for file management, execution, and connectivity. The main area is divided into two panels: a code editor on the left and a hardware diagram on the right.

Code Editor:

```
1 from microbit import *
2
3 while True:
4     display.
5
```

Hardware Diagram:

The hardware diagram shows a Micro:bit board with a grid of LEDs. The board is labeled with 'A' and 'B' at the bottom corners. The bottom edge shows pins labeled 0, 1, 2, 3V, and GND. Below the diagram are control buttons for play, pause, and reset.

Micro:bit , les autres capteurs

Un bouton de réinitialisation, désigné « R » pour « Reset »

Un accéléromètre pour détecter les mouvements de l'appareil (détection des actions comme secouer, pencher et la chute libre)

Une boussole magnétique 3D pour détecter les champs magnétiques : l'orientation de la carte.

Un capteur de température (sur le processeur)

Un capteur de luminosité lié aux LED

Une connectique bluetooth 4.0 basse énergie/2.4 GHz maître/esclave

Un connecteur micro-usb pour le programmer, connecteur alimentation 3,3V

