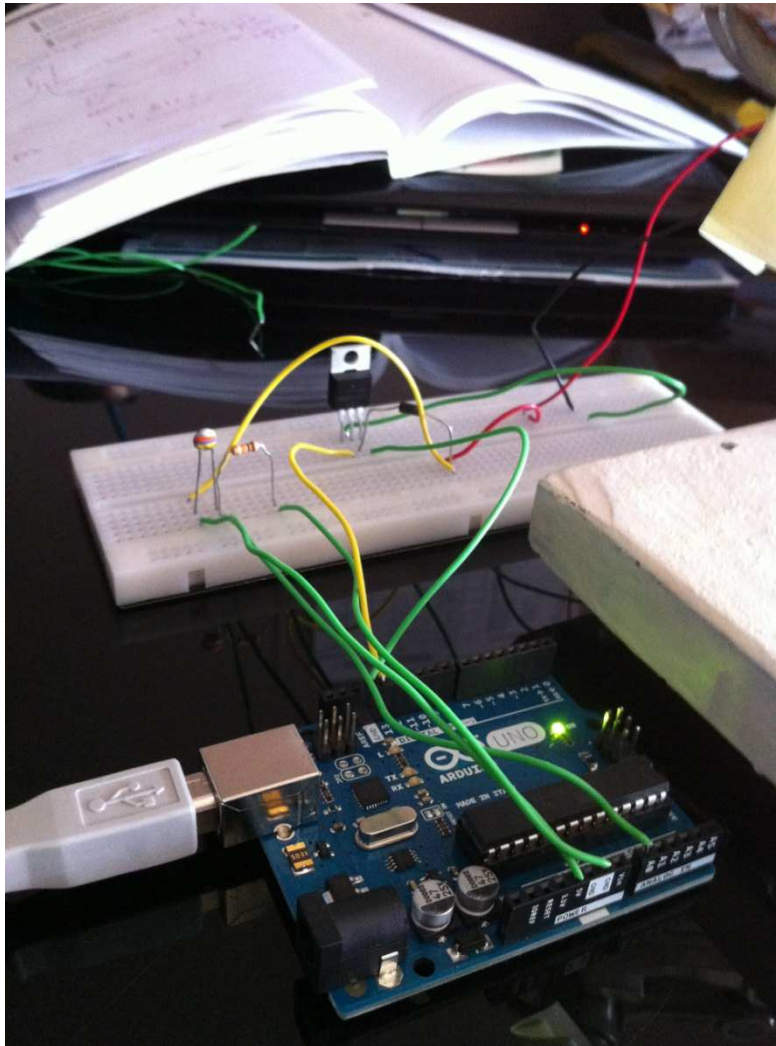


# Initiation à la démarche algorithmique et à la programmation informatique



## 1ère édition

Savoir faire :

- ❖ Concevoir et représenter un algorithme en vue de programmer le comportement d'un système technique.
- ❖ Utiliser un environnement de programmation graphique pour réaliser un programme commandant un système technique simple.

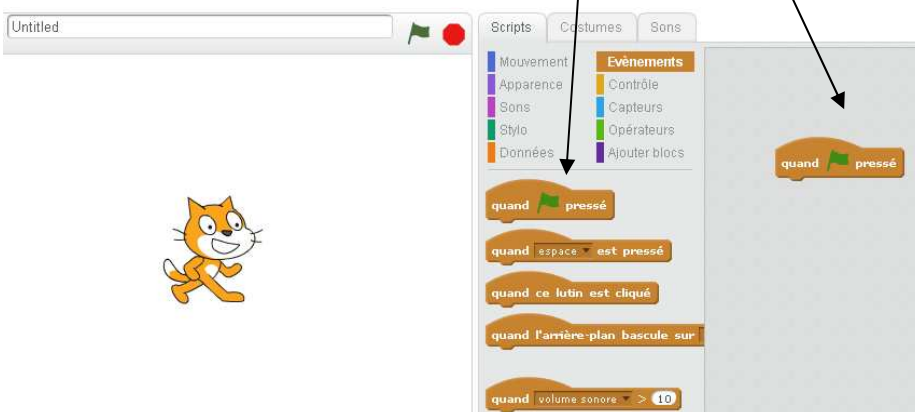
# Programmation avec le logiciel Scratch

Dans un premier temps, tu vas t'initier en ligne avec le logiciel Scratch.

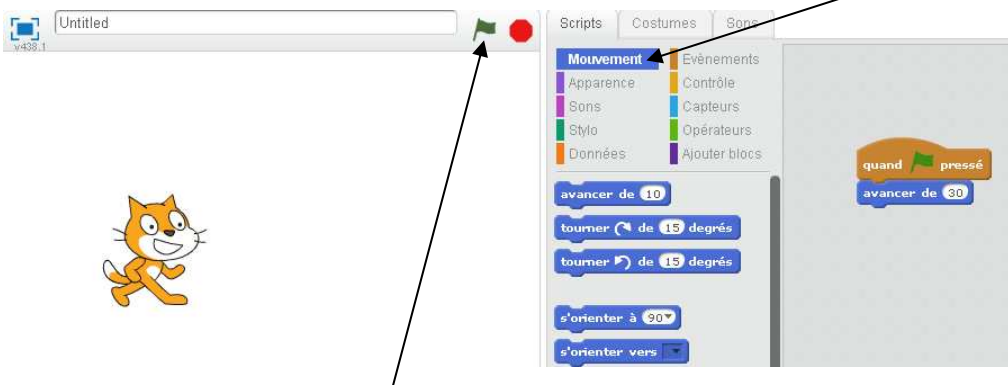


## 1. Programme simple :

- Rends-toi sur le site : <https://scratch.mit.edu/projects/editor/#editor>
- Tu vas réaliser le programme suivant :
- Clique sur « événement » puis sur l'icône et glisse-le à droite de l'écran.



- Puis tu vas faire avancer le chat de 30 pas. Pour cela clique sur « Mouvement » puis glisse le script sur la page de droite, puis modifie 10 par 30.

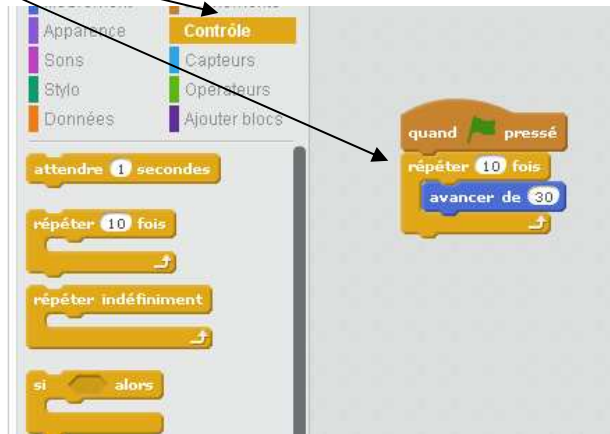


- Clique sur la flèche verte et tu vas observer le chat se déplacer.

## 2. Programme avec répétition d'exécution :

On va demander à la programmation de répéter plusieurs fois le programme.

- Clique sur « contrôle » puis glisse cet icône à la programmation en veillant à inclure le mouvement « avancer de 10 pas »



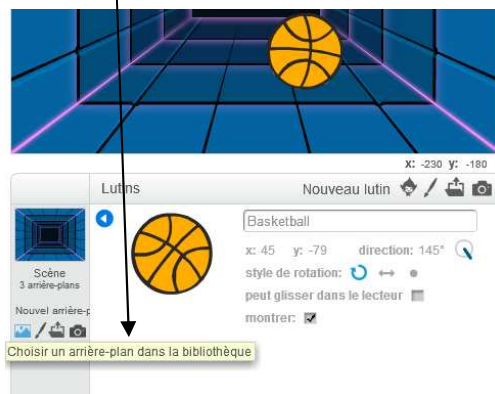
- Exécute ce programme en cliquant sur la flèche et tu vas t'apercevoir que le chat va avancer de 10 X 30 soit 300 pas.
- Vérifie par ce programme ci-dessous et tu verras qu'il aura parcouru la même distance.



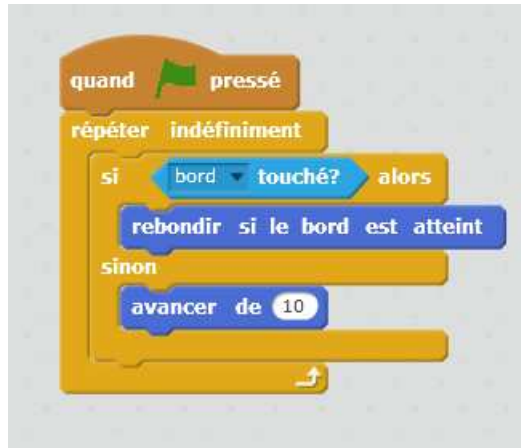
## 3. Programme avec condition logique « si alors sinon » :

Dans un premier temps, on va créer un nouveau fichier appelé balle rebondissante.

- Clique sur « choisir un arrière plan dans la bibliothèque, puis « neon tunnel ».



- Puis de la même manière choisis une balle de basket.
- Construis le programme suivant :



Le ballon va rebondir dans tous les sens.

- d. **Travail demandé** : ajoute un son à chaque fois qu'il touche le bord. Durée 20 minutes maxi.

Correction :



- e. Ajoute une cloche et imagine un programme tel que si le ballon touche la cloche alors Il ne rebondit plus.



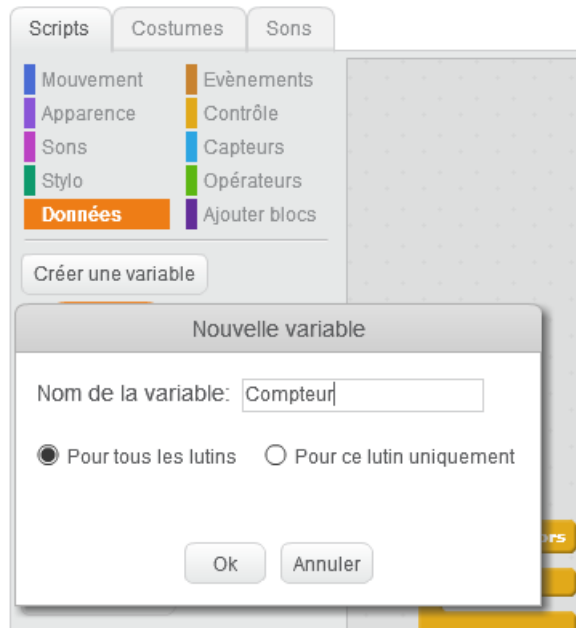
Correction :



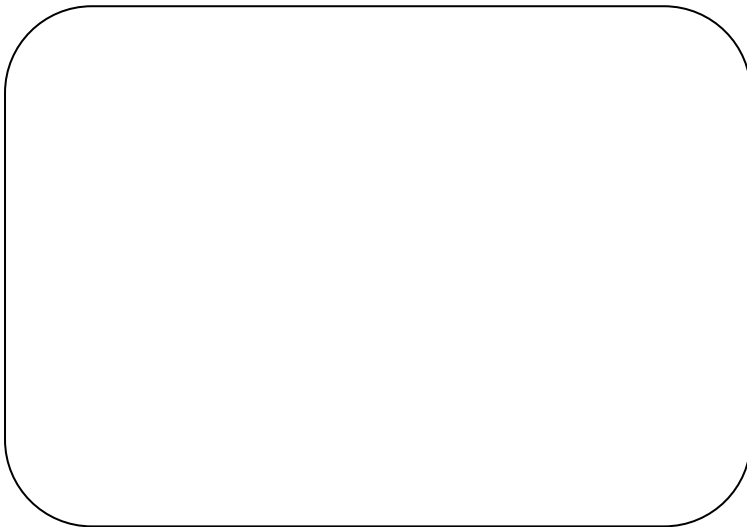
#### 4. Programme avec une variable, un compteur :

On va cette fois-ci créer une variable un compteur.

- Clique sur « Données ». Une fenêtre s'affiche.
- Tape le nom de la variable « compteur » puis ferme la fenêtre.



**Travail demandé** : à chaque fois que le ballon touche la cloche, un compteur va s'afficher et noter le nombre de touche. 1h.



Projet en groupe 2 semaines : avec tous ce que tu as appris, tu vas maintenant concevoir un projet que tu présenteras aux autres membres du groupe.

# Démarche algorithmique avec Algobox

## 1. Présentation du logiciel :

Nous avons vu les démarches algorithmiques avec le langage Arduino. Nous avons défini comment définir une variable. `int val = 0` permet de définir une variable au programme appelé `val` et qui sera un nombre entier grâce à l'abréviation `int`. `=0` veut dire que l'on va attribuer au départ à cette variable le chiffre zéro.

De la même manière avec Algobox, nous devons définir les variables, que l'on va stocker dans la mémoire... On peut reprendre le premier exemple du tutoriel où l'on demande de concevoir un programme qui puisse convertir des francs en euros.

Durée : 20 minutes

- a. Tout d'abord, il va falloir stocker quelque part le prix en francs (entré par l'utilisateur) et le résultat (le prix en euros). Pour cela on utilise ce qu'on appelle en informatique des **variables**. Ici, nos deux variables stockeront des nombres et nous les appelleront `prixenfrancs` et `prixeneuros`.  
La première chose à faire dans notre algorithme va donc être d'expliquer à l'ordinateur qu'il va falloir utiliser deux variables (contenant des nombres) appelées `prixenfrancs` et `prixeneuros`.  
Voilà comment faire avec **AlgoBox** (lancer d'abord le programme, si ce n'est pas déjà fait):
  - o Cliquer sur le bouton Déclarer nouvelle variable .
  - o Dans le champ Nom de la variable, entrer `prixenfrancs`, vérifier que le Type de la variable est bien sur **NOMBRE** et cliquer sur OK .
  - o Cliquer de nouveau sur le bouton Déclarer nouvelle variable.
  - o Dans le champ Nom de la variable, entrer `prixeneuros`, vérifier que le Type de la variable est bien sur **NOMBRE** et cliquer sur OK .
- b. Il faut maintenant permettre à l'utilisateur d'indiquer le prix en francs qu'il veut convertir et stocker ce nombre dans la variable `prixenfrancs`.  
Voilà comment faire avec **AlgoBox** :
  - o Avec la souris, se placer (si ce n'est pas déjà le cas) sur la ligne `DEBUT_ALGORITHMME`, puis cliquer sur le bouton Nouvelle Ligne.
  - o Cliquer alors sur le bouton Ajouter LIRE variable.
  - o Dans le champ LIRE la variable, vérifier que `prixenfrancs` est bien sélectionné et cliquer sur OK.
- c. Procédons maintenant au calcul du prix en euros (rappel : 1 euro représente 6.55957 francs).  
Voilà comment faire avec **AlgoBox** :
  - o Avec la souris, se placer (si ce n'est pas déjà le cas) sur la ligne `LIRE prixenfrancs`, puis cliquer sur le bouton Nouvelle Ligne.
  - o Cliquer alors sur le bouton AFFECTER valeur à variable.
  - o Sélectionner `prixeneuros` juste après La variable, taper `prixenfrancs/6.55957` dans le champ prend la valeur et cliquer sur OK (Attention à bien taper 6.55957 et pas 6,55957 : en informatique, le séparateur décimal est le point et pas la virgule).
- d. Il ne reste plus qu'à afficher le résultat.  
Voilà comment faire avec **AlgoBox** :
  - o Avec la souris, se placer (si ce n'est pas déjà le cas) sur la ligne `prixeneuros PREND_LA_VALEUR prixenfrancs/6.55957`, puis cliquer sur le bouton Nouvelle Ligne.
  - o Cliquer alors sur le bouton Ajouter AFFICHER variable.
  - o Sélectionner `prixeneuros` juste après AFFICHER la variable et cliquer sur OK.

e. Notre algorithme est maintenant terminé et devrait correspondre à ça :

```
▼ VARIABLES
|
|  └─ prixenfrancs EST_DU_TYPE NOMBRE
|  └─ prixeneuros EST_DU_TYPE NOMBRE
▼ DEBUT_ALGORITHME
|
|  └─ LIRE prixenfrancs
|  └─ prixeneuros PREND_LA_VALEUR prixenfrancs/6.55957
|  └─ AFFICHER prixeneuros
└─ FIN_ALGORITHME
```

Il n'y a plus qu'à le tester :

- o Cliquer sur le bouton Tester Algorithme.
- o Dans la fenêtre qui s'affiche, cliquer alors sur le bouton Lancer algorithme.
- o Entrer alors un prix à convertir (100 par exemple) et cliquer sur OK. Le prix en euros apparaît alors dans le cadre Résultat .
- o Pour convertir un autre montant, il suffit de cliquer à nouveau sur le bouton Lancer algorithme.
- o Une fois les tests terminés, on revient à la fenêtre principale du programme en cliquant sur le bouton fermer en bas de la fenêtre. On peut alors sauvegarder son algorithme en cliquant sur le bouton Sauver.

Source : <http://www.xm1math.net/algobox/tutoalgobox/page1.html>

## 2. Exercice d'application :

On décide d'organiser un voyage en Angleterre. La monnaie locale est la livre sterling.

1 euros = 0,69851 livre sterling.

Faire le programme et l'afficher. Durée 20 minutes.

### **3. Programmation avec condition logique « si....alors.....sinon » :**

Je te demande de créer un programme tel qui puisse calculer l'inverse d'une valeur.

X est la variable. Attention je te rappelle que l'inverse de zéro n'existe pas. Durée 1h.

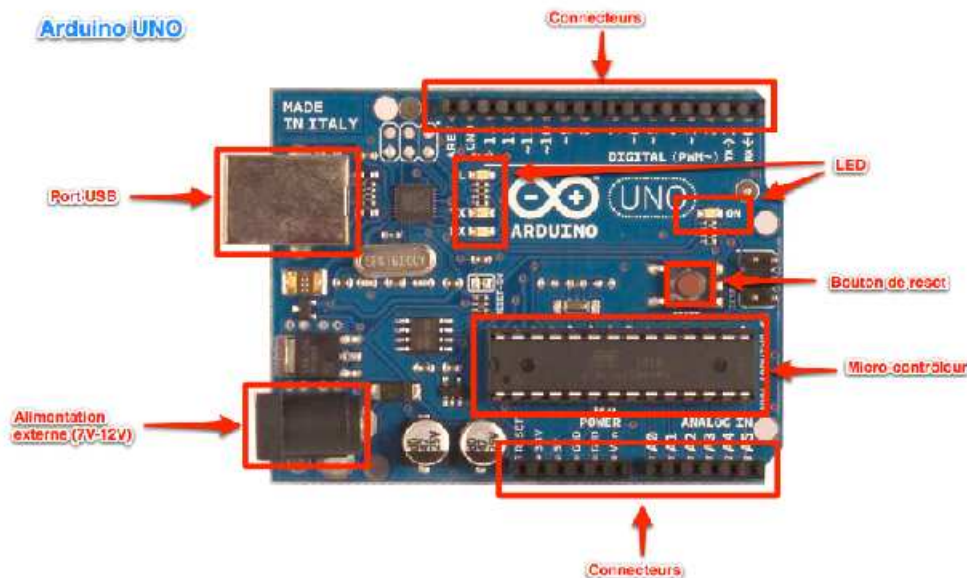


# Programmation avec Arduino

## 1. Présentation de la carte Arduino :

L'Arduino est une plateforme open source d'électronique programmée qui est basée sur une carte à microcontrôleur et un logiciel. Plus simplement, on peut dire que l'Arduino est un module électronique, doté d'un microcontrôleur programmable.

Il peut être utilisé pour développer des objets interactifs, munis d'interrupteurs ou de capteurs, et peut contrôler une grande variété de lumières, moteurs ou toutes autres sorties matérielles.



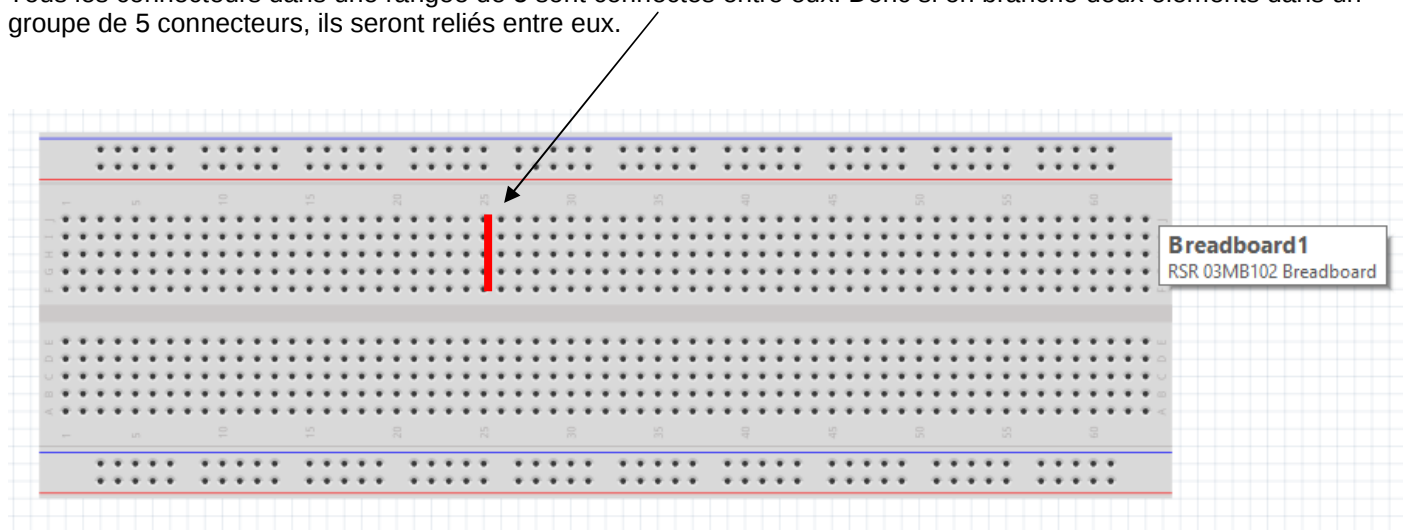
Images : Diduino2014.pdf

## 2. Présentation de la platine de connexion :

Une platine d'expérimentation permet de réaliser des prototypes de montages électroniques sans soudure et donc de pouvoir réutiliser les composants.

Elle permet de connecter les composants entre eux et de créer des circuits. Il y a 17 rangées. Dans chaque rangée, il y a deux groupes de 5 colonnes, séparés par une tranchée au milieu.

Tous les connecteurs dans une rangée de 5 sont connectés entre eux. Donc si on branche deux éléments dans un groupe de 5 connecteurs, ils seront reliés entre eux.



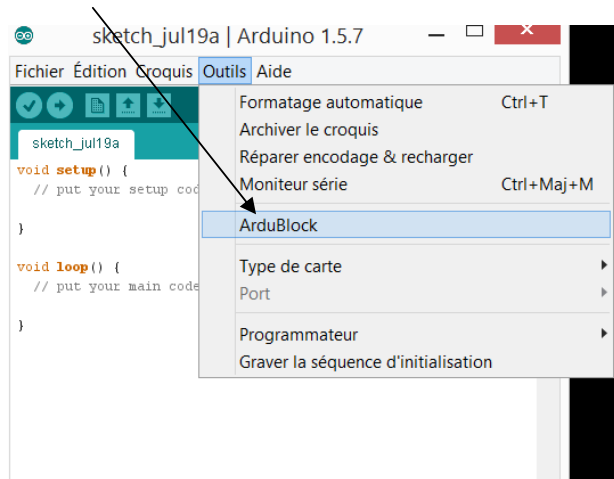
### 3. Principe de fonctionnement :

Il est simple :

1. On réalise le programme sur un ordinateur.
2. On connecte l'ordinateur à l'Arduino via une prise USB.
3. On envoie le programme sur l'Arduino.
4. L'Arduino exécute enfin le programme de manière autonome.

On va utiliser un logiciel ArduBlock qui permet de programmer en bloc de fonction comme Skatch.

Installe-le en utilisant les tutoriels sur Internet. En lançant Arduino tu dois le visualiser dans la barre de menu « outils »



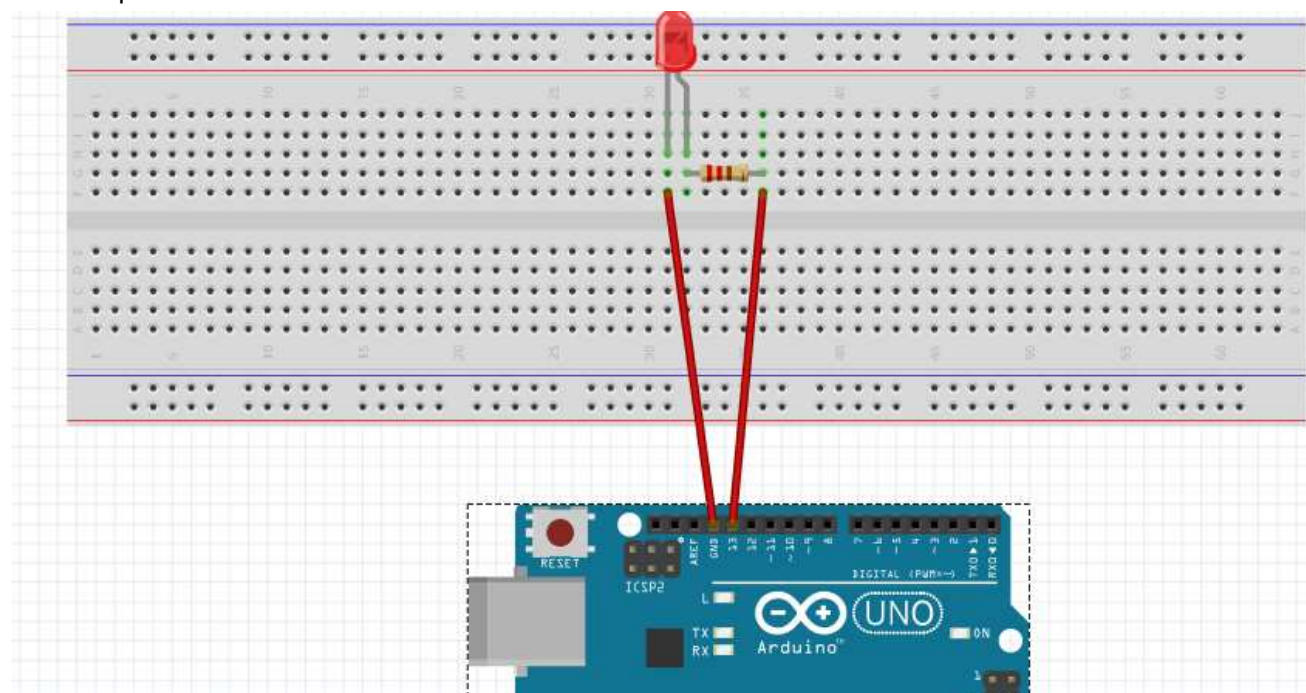
### 4. Travail demandé :

Tu vas réaliser le montage suivant :

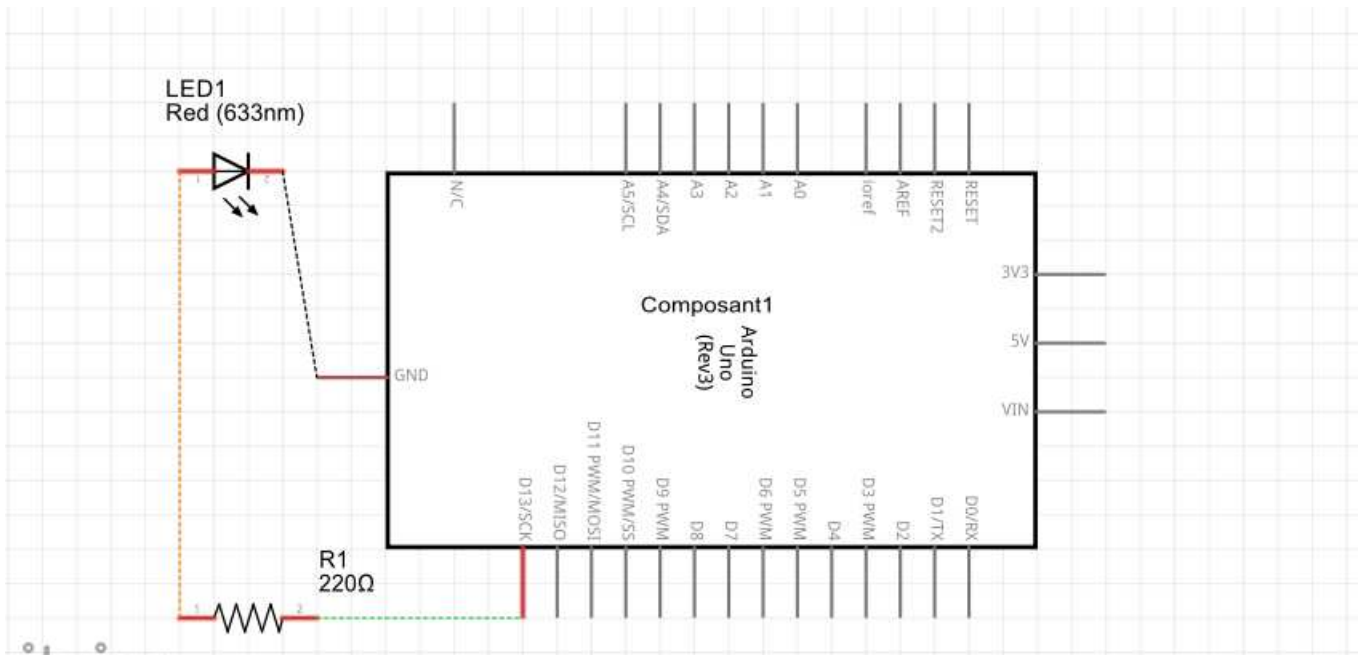
Matériels : Une DEL, une résistance de protection 220  $\Omega$ . Un fil noir (-) et un fil rouge (+).

- D'abord avec un logiciel de schéma électronique que tu auras téléchargé sur : <http://fritzing.org>

Voici ce que tu dois dessiner :



Tu cliques sur « vue schématique » et tu vois le schéma structurel.



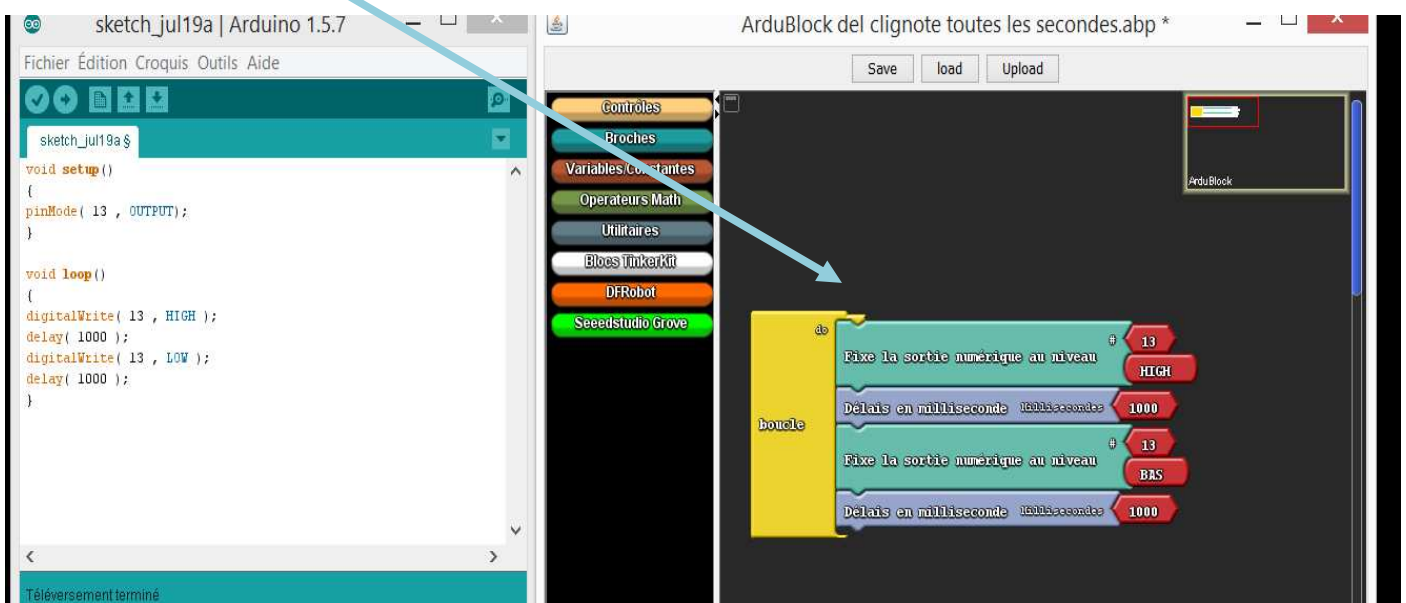
- Ensuite réalise le montage en n'oubliant pas de connecter le fil rouge sur la sortie numérique 13 et le fil noir sur GND (la masse).



Tu ne connectes pas la carte Arduino à l'ordinateur sans l'accord du professeur qui aura validé ton montage.

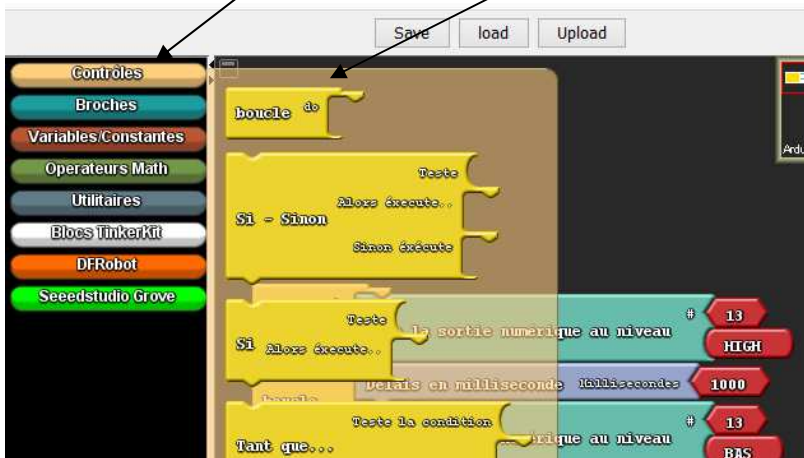
- Ensuite on va sur Ardublock. On va créer un programme qui va nous permettre de faire clignoter la DEL tous les secondes.

Voici le programme Ardublock :

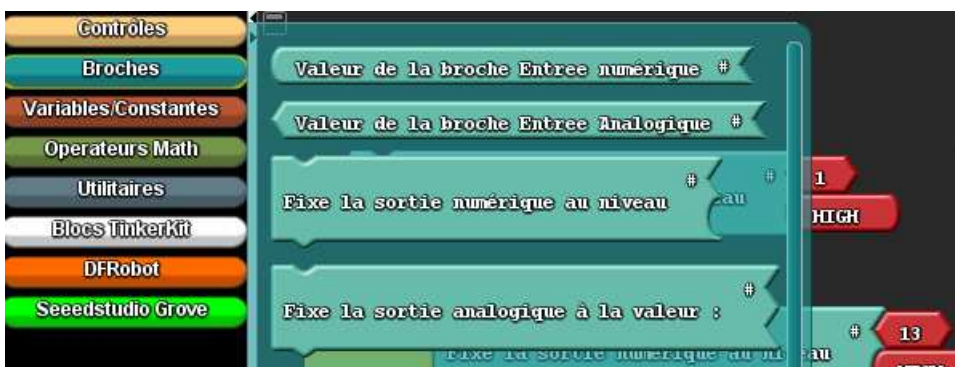


## 5. Programmation avec Ardublock :

- a. Clique d'abord sur « Contrôles » puis sélectionne « boucle de » et fais le glisser dans la page de droite.



- b. Puis clique sur « Broches » et fixe « la valeur de la sortie numérique ».  
c. Change la valeur 1 par 13 et vérifie que tu as bien sélectionné « Haut ».



- d. Clique ensuite sur « Utilitaires » et sélectionne « Delais en millisecondes ».



e. Etablis ce programme :



f. Puis clique sur « Upload » pour le télécharger sur la carte.



Tu ne connectes pas la carte Arduino à l'ordinateur sans l'accord du professeur qui aura validé ton montage.

g. Puis sur Arduino, tu obtiens automatiquement le programme suivant :

```
Fichier Édition Croquis Outils Aide
sketch_jul19a $
void setup()
{
  pinMode( 13 , OUTPUT);
}

void loop()
{
  digitalWrite( 13 , HIGH );
  delay( 1000 );
  digitalWrite( 13 , LOW );
  delay( 1000 );
}
```

Téléverser le programme dans la carte

Vérifie si le programme ne présente pas d'erreurs

Met la broche en sortie

Programme en boucle :

1. Allume la DEL.
2. Attendre 1 seconde.
3. Eteint la DEL.
4. Attendre 1 seconde

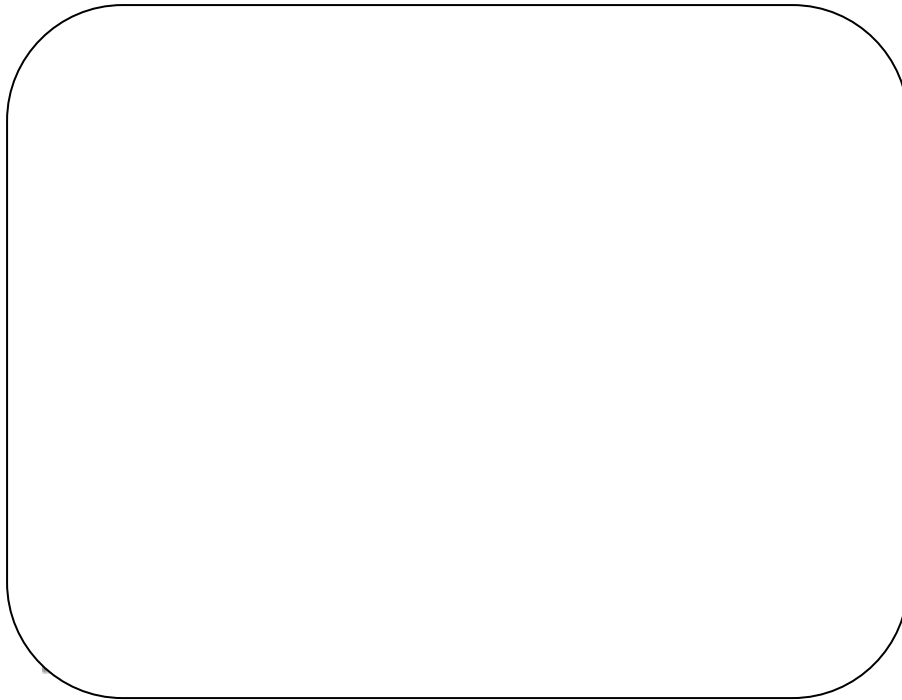
Remarque : cela te rappelle le même principe de programmation avec Scratch. On y reviendra pus tard.

## 6. Exercice d'entraînement :

Tu vas réaliser un montage où cette fois-ci, tu vas pouvoir faire clignoter 2 del (comme dans une voiture qui met ces warnings). Tu devras mobiliser tes connaissances en physique chimie.

**Attention une condition doit être respectée** : si tu enlèves une des deux dels, l'autre devra encore clignoter.

Correction :



### 7. Programmation avec condition logique « si....alors.....sinon » :

Nous avons vu avec Skrachth comment programmer avec une condition logique. Nous allons réaliser un même programme avec Arduino en utilisant un capteur de luminosité.

Résolution de problème : nous voulons allumer une del automatiquement dès que la pièce s'assombrit.

Comment faire ?

Construis l'algorithme par des phrases.

Au départ, on mesure la luminosité de la pièce :

Cœur du programme : S'il fait jour, alors ne pas allumer la del.

Sinon s'il fait nuit, allumer la del.

Dans un premier temps, il faut définir une variable : val pour enregistrer quel est l'état de l'entrée du capteur analogique A0. Ce n'est ni plus ni moins une case mémoire de la carte dans laquelle on peut stocker des données. Int val =0 signifie que l'on va stocker un nombre entier : val est le nom de la variable et =0 permet de lui assigner le nombre 0 à cette variable au départ.

Attention chaque ligne se termine par un point virgule. Cela signifie qu'une ligne vient de se terminer.

**Il faut bien lui dire que la lecture de val se trouve sur l'entrée analogique A0 sur ta carte Arduino.**

**Exercice demandé :** construis le programme sur Ardublock Durée 1h.



Tu ne connectes pas la carte Arduino à l'ordinateur sans l'accord du professeur qui aura validé ton montage.

Correction :

**Ensuite construis le programme Arduino. 1/2h.**

Indication : voici le début du programme et les explications après les deux // :

```
int val=0; //sauve la mesure du capteur
void setup()
{
  pinMode( 13 , OUTPUT); //DEL connecté sur la sortie numérique 13 de la carte Arduino.
  pinMode( val , INPUT ); // Le capteur de luminosité est connecté sur l'entrée analogique A0. Pas obligé de l'écrire...
}
void loop() {
  val = analogRead (0); //Mesure la valeur du capteur connecté à A0. Ici on écrit 0 pour définir A0.
  if.....
```

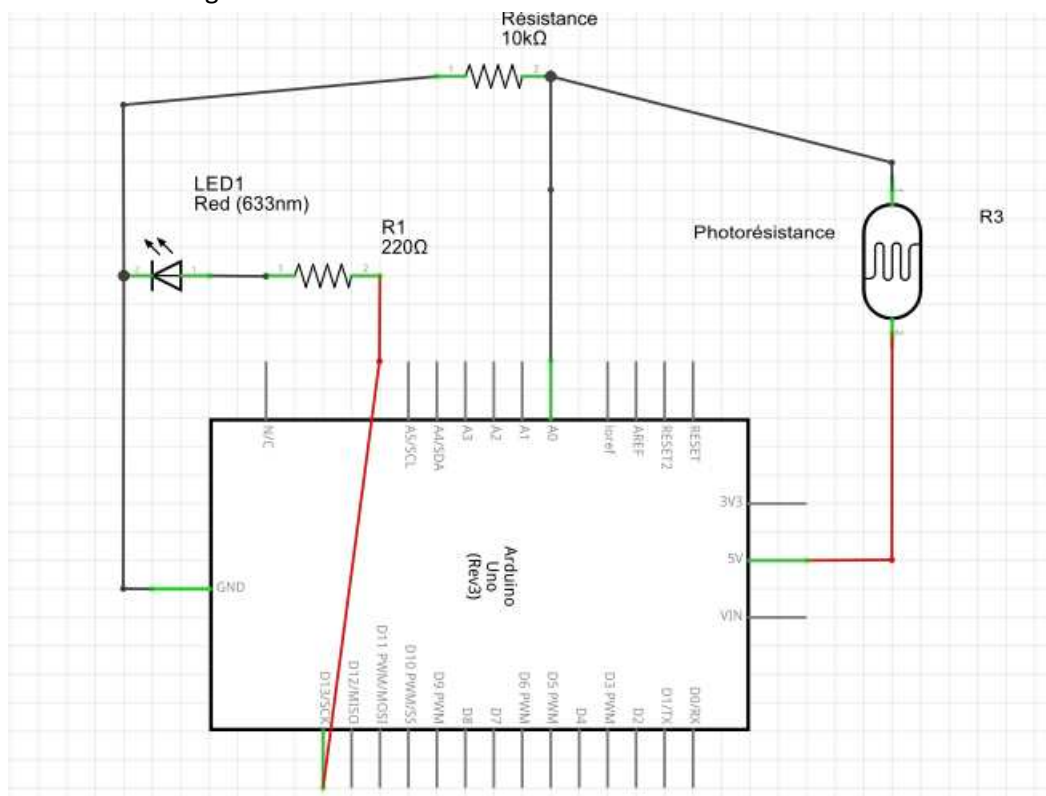
## Correction:

Matériels :

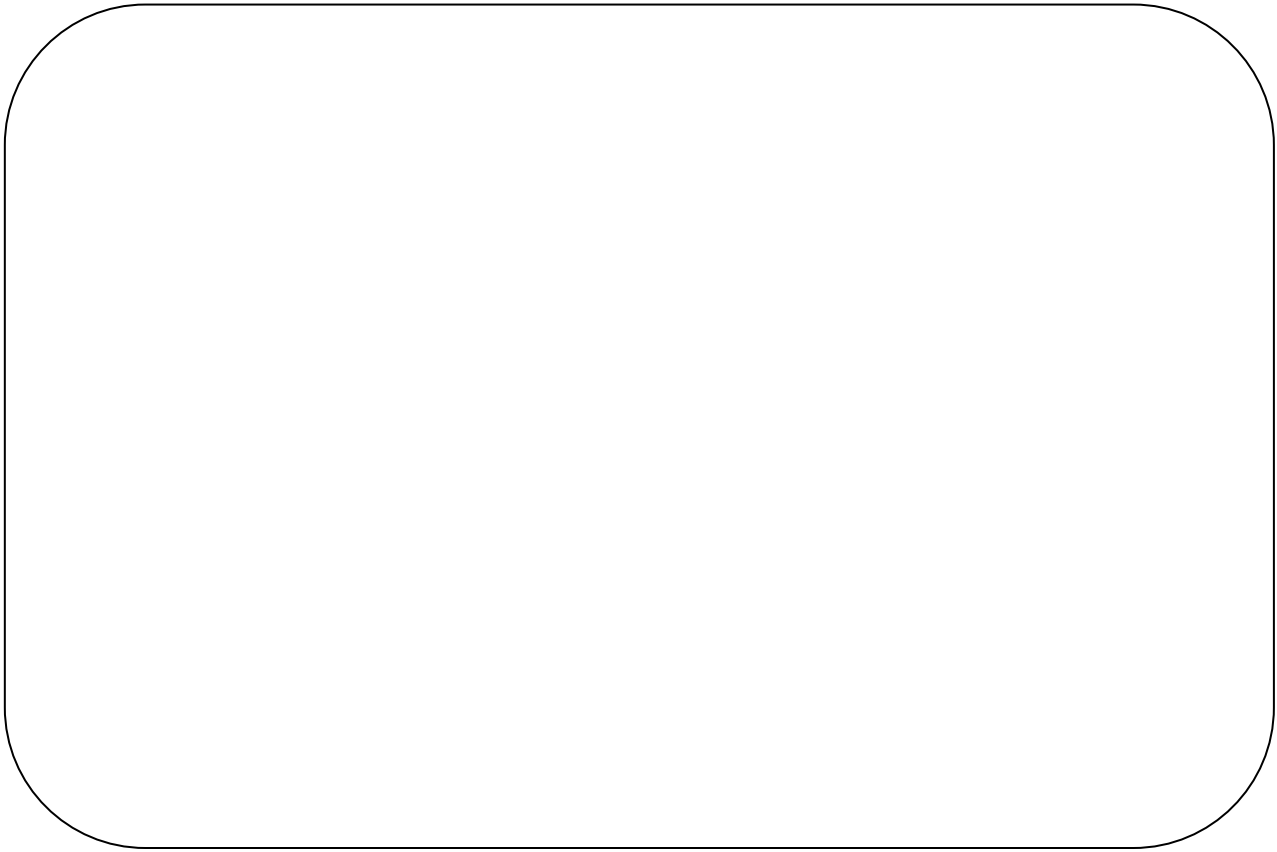
- ❖ 1 résistance de 10K $\Omega$  reliée à la photo résistance.
- ❖ 1 résistance de 220 $\Omega$  reliée à la photo résistance reliée à la led.
- ❖ 5 fils de connexion.
- ❖ La carte Arduino.
- ❖ 1 led rouge.

## 8. Montage.

Réalise le montage avec le carte. Durée 20 minutes.



Correction :



#### 9. Projet à réaliser : 2 semaines.

Tu vas concevoir un projet innovant avec ton groupe. Tu pourras utiliser un capteur de température pour refroidir ou chauffer une pièce.....Un détecteur de présence.....