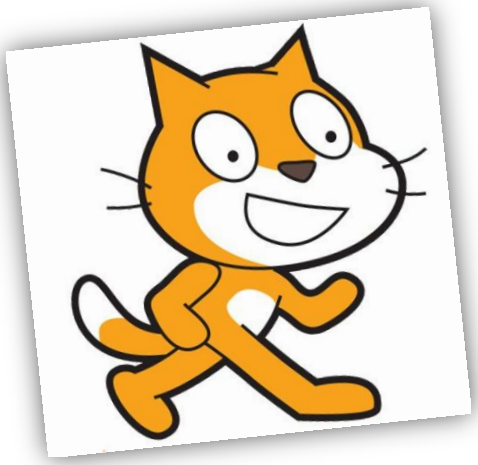
Académie de Poitiers

Enseigner
la programmation et l'algorithmique
au COLLÈGE avec

SCRATCH



LIVRET N°1

(aspects techniques et pratiques de l'application)

2015-2016

Sommaire

Présentation de SCRATCH

Scratch est un langage de programmation visuel et gratuit. En glissant-déposant des blocs colorés, vous pouvez créer des histoires interactives, des jeux, des animations, de la musique, ou des présentations. Il est ensuite possible de les télécharger sur Internet pour les partager avec les autres utilisateurs du monde entier. Scratch est conçu pour jouer, apprendre par soi-même et créer.

Il est conçu par le **MIT** pour initier les jeunes à des concepts importants en mathématiques et informatique, tout en apprenant à développer une pensée créative, un raisonnement systématique et à travailler en équipe.

Vous allez utiliser des **« briques »** qui contiennent des **commandes** et **peuvent s'emboîter** ensemble **pour créer des programmes**. Ces programmes dirigent les personnages et les objets dans le jeu.

Un "objet" est un personnage ou un objet de votre jeu. Les objets peuvent se déplacer et être actif ou être des accessoires inactifs.
Les objets ne peuvent rien faire par eux-mêmes. L'action d'un **objet** vient des **scripts** de la **fenêtre de scripts**. Les zones de scripts sont propres à chaque objet. Ces scripts sont des instructions pour dire au lutin ce qu'il doit faire.

Les **instructions** sont des ensembles de **briques**. Vous faites glisser ces briques depuis la zone des briques vers la zone "Scripts". Ces briques s'emboîtent comme un puzzle pour créer les instructions

D'où vient le nom Scratch ?

Le *scratching* est une technique utilisée par les DJ pour mixer des morceaux de musique. De la même manière, la programmation avec Scratch associe différents objets (images, photos, effets sonores, ...) pour réaliser quelques chose d'entièrement nouveau.

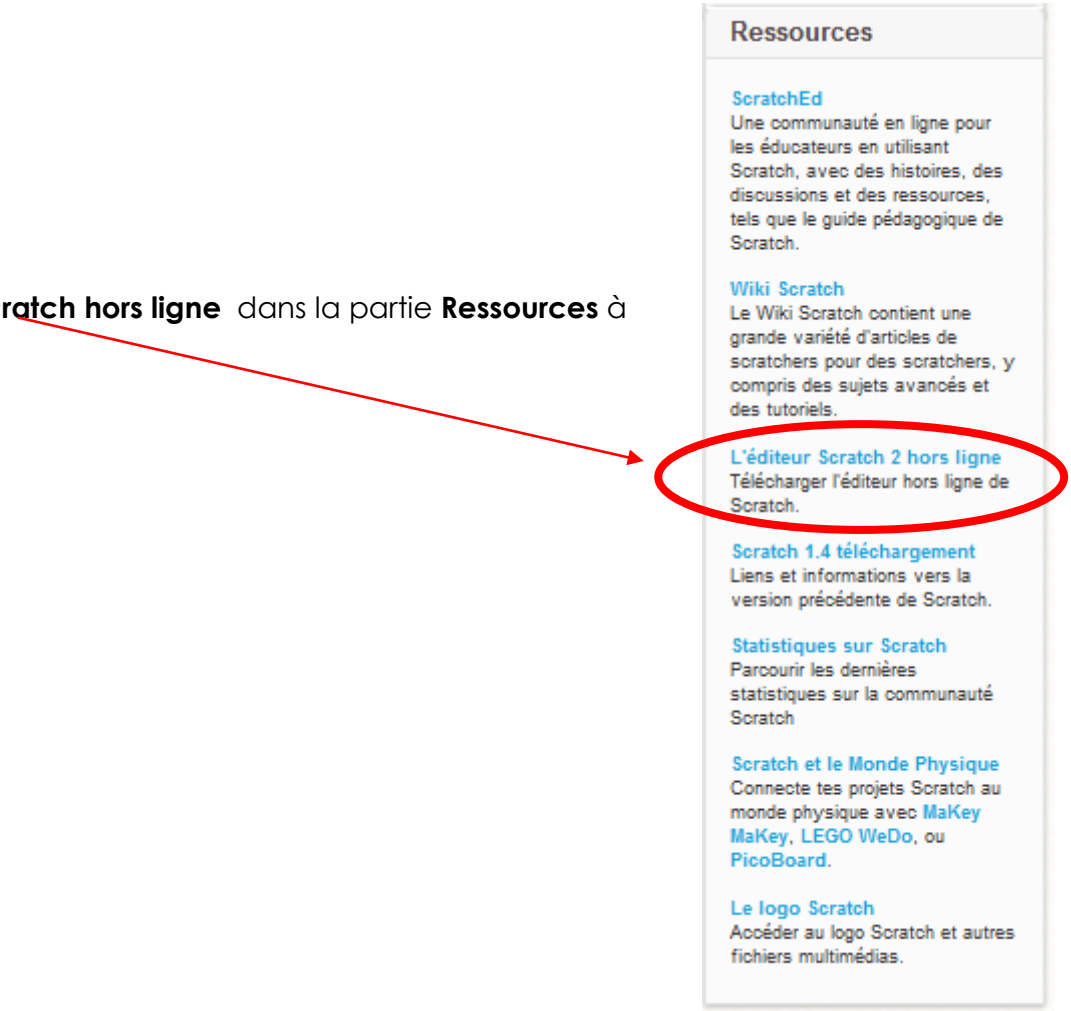
Télécharger la version **HORS LIGNE** de **SCRATCH 2.0**

Tutoriel

1. Se rendre sur le site **https://scratch.mit.edu/**

2. Cliquer sur **Aide** (menu en haut de la page d'accueil du site)

3. Cliquer sur **L'éditeur de Scratch hors ligne** dans la partie **Ressources** à droite de la page



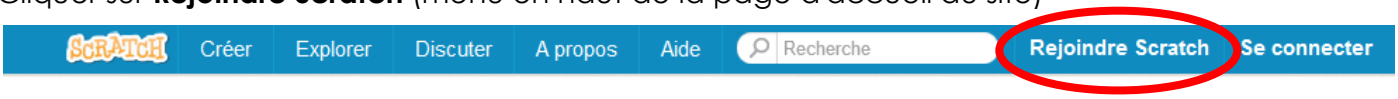
4. Cliquer sur **télécharger** (en choisissant le système d'exploitation correspondant à votre ordinateur) et suivre les instructions de téléchargement

Utiliser la version **EN LIGNE** de **SCRATCH 2.0**

Tutoriel

1. Se rendre sur le site **https://scratch.mit.edu/**

2. Cliquer sur **Rejoindre Scratch** (menu en haut de la page d'accueil du site)



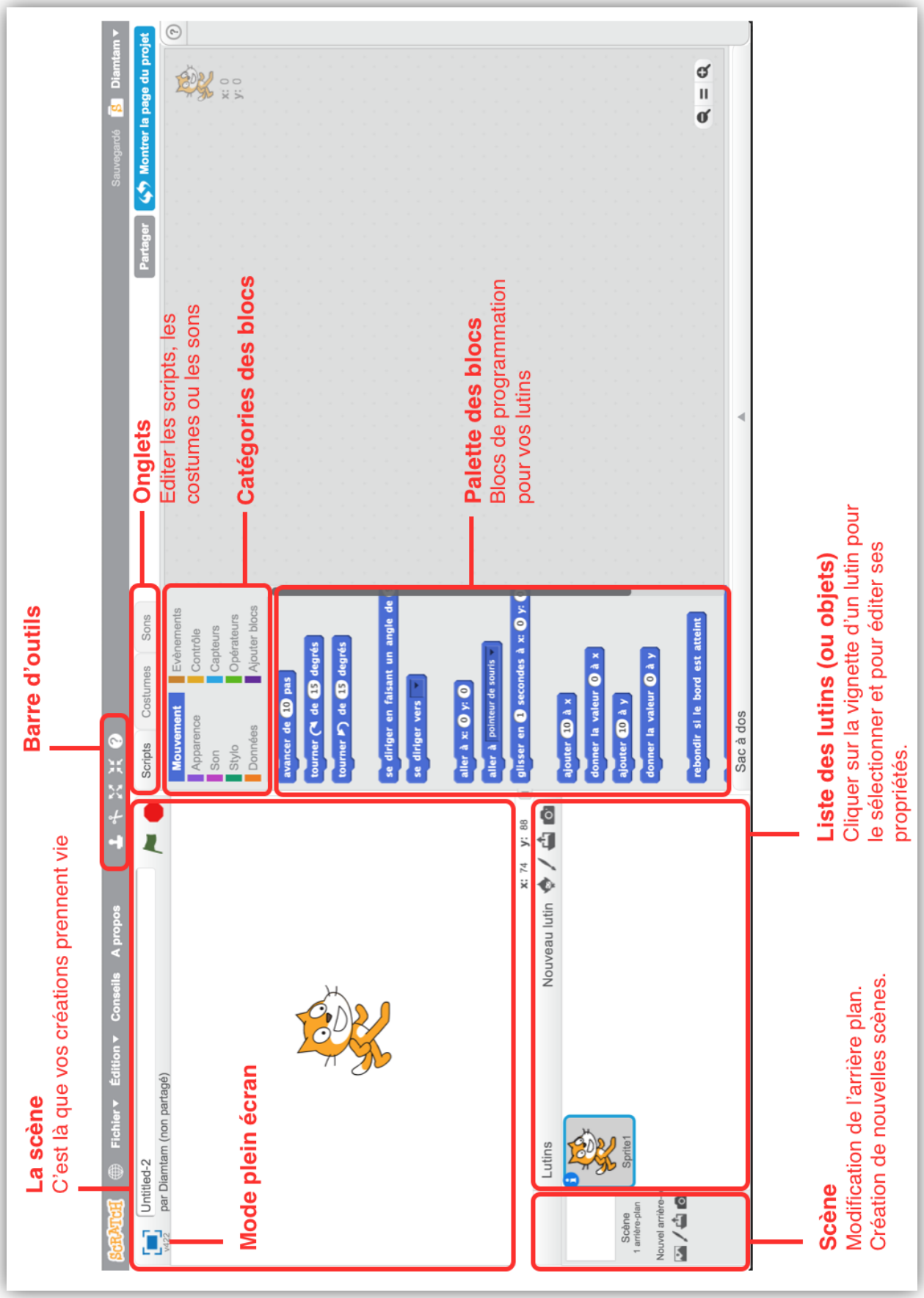
Remarque :
Il est possible de créer des programmes sur Scratch, sans s'inscrire.
L'avantage de s'inscrire est de pouvoir conserver en ligne l'ensemble de ses projets et de partager avec la communauté Scratch.

3. **Suivre** les étapes d'inscription

4. Ensuite cliquer sur **Créer.** C'est parti, vous pouvez commencer un nouveau projet.

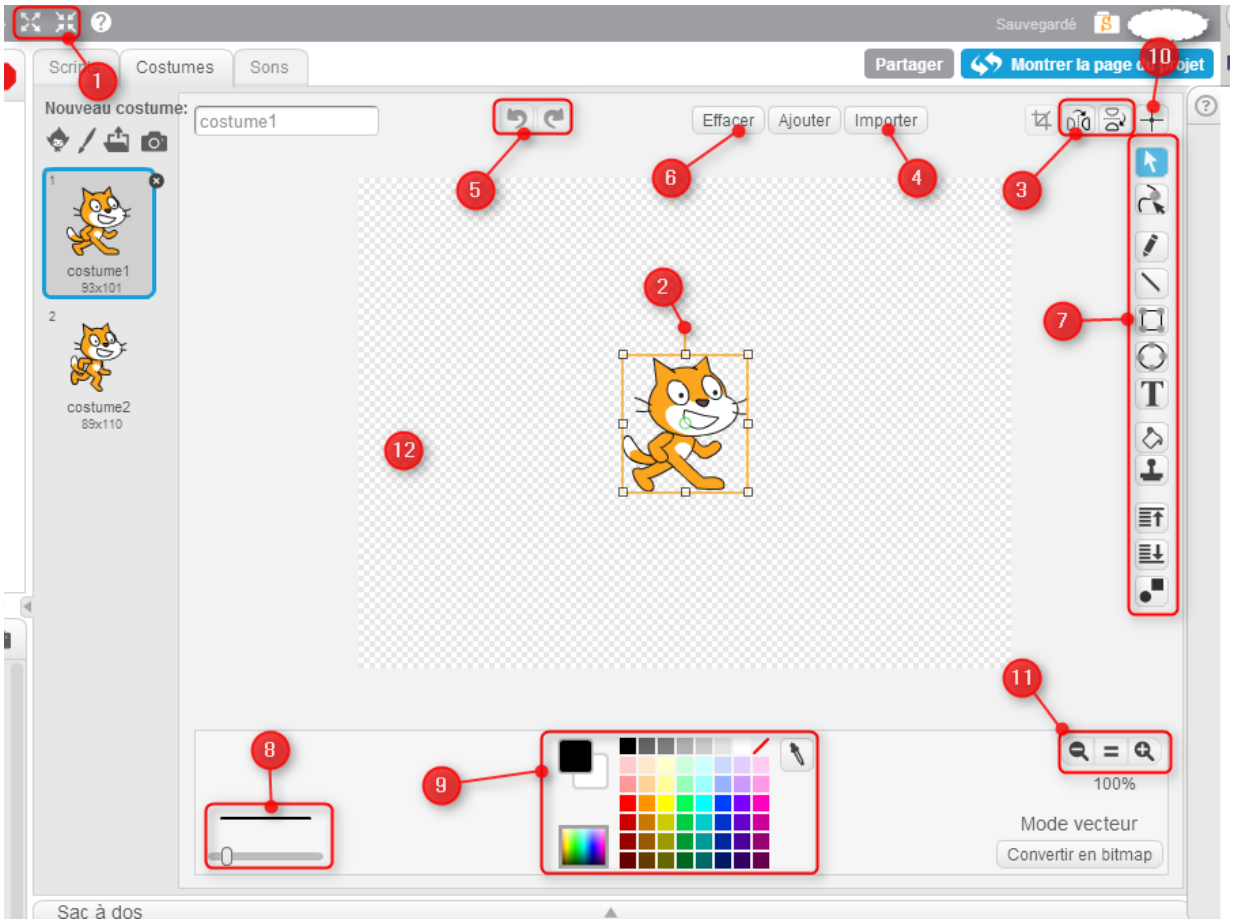
Remarque :
*Il est possible de consulter les projets partagés sur Scratch en cliquant sur **Explorer**.*

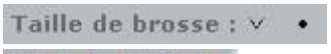
L'interface de Scratch 2.0



source : http://www.epslemont.ch/

Pour changer le costume du lutin



- **Zone 1**  : agrandir ou réduire votre dessin.
- **Zone 2**  : effectuer une rotation à votre dessin.
- **Zone 3**  : retourner votre dessin.
- **Zone 4**  : importer une image.
- **Zone 5**  : annuler ou rétablir une action.
- **Zone 6**  : effacer votre dessin.
- **Zone 7**  : palette d'outil de dessin.
- **Zone 8**  : taille du crayon à dessin.
- **Zone 9**  : palette de couleurs.
- **Zone 10**  : définir l'axe de rotation de votre dessin.
- **Zone 11**  : zoom.
- **Zone 12** : zone de dessin.

Présentation des principaux éléments du langage Scratch

Introduction

Cette présentation a pour objectif de vous faire découvrir en quelques pages les principaux éléments du langage **Scratch** tout en faisant ressortir la spécificité ce type de langage de programmation qui au premier abord peut être déroutant si avez déjà programmé et ne connaissez que des langages de programmation non orientée objet et non événementielle.
Mais **Scratch** et conçu pour les enfants qui ne se posent pas les mêmes questions que vous, et l'aspect ludique de **Scratch** avec son système d'assemblage des briques facilite l'initiation à ce type de programmation sans avoir de connaissances préalables en programmation, alors pas de panique !

L'objet scène

- La **scène** est une partie de l'écran, que l'on peut mettre en plein écran, sur laquelle le programme créé affiche, fait évoluer et interagir les différents objets graphiques (**lutins**).
- La **scène** est munie d'un système de coordonnées entières x et y dont l'origine est le centre de la **scène**, et tel que $-240 \leq x \leq 240$ et $-180 \leq y \leq 180$.
- La **scène** peut avoir plusieurs **arrière-plans** qui sont chargés depuis la bibliothèque ou un fichier, ou bien créés à l'aide de l'**éditeur graphique** ou la webcam. Le programme (ensemble de différents **scripts**) pourra changer l'**arrière-plan** de la **scène**.
- La **scène** possède sa propre **fenêtre de scripts**.

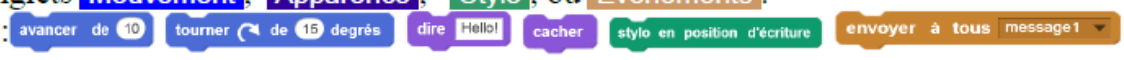
Les objets graphiques (lutins)

- Un **lutin** est une image que l'on peut charger depuis la bibliothèque ou un fichier. On peut aussi créer un lutin avec l'**éditeur graphique** ou la webcam.
- On peut créer plusieurs **costumes** pour un même **lutin** et le programme pourra ainsi changer l'apparence de chaque **lutin** créé s'il a plusieurs **costumes**.
- Chaque **costume** possède un **centre** que l'on peut modifier dans l'éditeur graphique à l'aide du bouton ⊕. Ce **centre** (*qui n'est pas nécessairement au centre de l'image*) est associé aux coordonnées du **lutin**, c'est autour de ce **centre** que le **lutin** pourra tourner sur lui même, et c'est ce **centre** qui laissera une trace sur la **scène** lors des déplacements du **lutin** si son **stylo** est abaissé.

Les scripts

- Chaque **lutin** possède sa propre **fenêtre de scripts**.
- Un **script** est une pile d'instructions qui s'exécute de haut en bas qui commence par un écouteur d'événements.
  Les **scripts Scratch** s'écrivent en empilant les **blocs de commande** sous un premier **bloc chapeau** (attente d'un événement précis) accessible dans l'onglet Evenements
  Ci-dessous les différents **blocs chapeaux** :

- Dès sa création un **lutin** est créé avec ses propres propriétés et **variables** correspondantes (abscisse, ordonnée, direction, intensité et couleur d'écriture) et ses propres méthodes qui permettent de changer ses propriétés (méthodes de déplacement, de changement d'apparence, d'écriture, et de réponses aux événements) ou de communiquer avec les autres **lutins**.
- Les méthodes et **variables** d'un **lutin** ne peuvent être utilisées que dans les **scripts** de sa propre **fenêtre de scripts**. Elles correspondent à des **blocs de commande** que l'on trouve dans les onglets Mouvement, Apparence, Stylo, ou Evenements.
  Exemples :



- Un **script** ne peut pas accéder directement aux méthodes et variables d'un **lutin** s'il n'est pas sur la **fenêtre des scripts** de ce **lutin**.

Les événements

- Un **script** associé à un **lutin** ou à la **scène** n'est exécuté qu'à partir du moment où l'**événement** attendu précisé dans son **bloc chapeau** est déclenché par l'utilisateur (*clic sur le drapeau vert, clic sur un lutin, touche pressée*) ou lorsqu'un **message** est envoyé par un **script** de l'un des **lutins** ou de la **scène**.
- Les blocs 

ou 

permettent à la **scène** ou à un **lutin** d'envoyer des **messages** (*chaîne de caractères à écrire à la place de* 

) à tous les **lutins** et à la **scène**.
Si le message message1 est envoyé par un **script**, tous les **scripts** de tous les **lutins** ou de la **scène** commençant par le **bloc** 

seront alors exécutés.
- Voici un exemple de **script** qui créé un **événement** en envoyant un message dés que les touches a et z sont pressées.

Les variables prédéfinies

- Certaines **variables** sont prédéfinies même si aucun **lutin** n'est utilisé. Les **blocs de commande** associés à ces variables sont les suivants :

nom de l'arrière-plan	arrière-plan n°	volume	tempo	réponse	souris x
souris y	volume sonore	chronomètre	actuel minute	jours depuis 2000	nom d'utilisateur

- Les dix **blocs de commande** suivants correspondent à des **variables** propres à chaque **lutin**, elles sont relatives à des propriétés du **lutin**.

abscisse x du centre du lutin	ordonnée y du centre du lutin	direction de déplacement	costume n°	taille % de taille initiale	distance de à un autre lutin

Les quatre **variables** booléennes (valeur : **true** ou **false**) associées aux **blocs** ci-dessous ne sont pas modifiables par **script**. Elles sont actualisées à chaque mouvement du **lutin** et utilisées dans des instructions conditionnelles.

touché? Valeur true si le lutin touche l'autre lutin (ou bord, ou pointeur) sélectionné.	couleur touche ? Valeur true si la 1ere couleur (du lutin) touche la 2eme couleur.	couleur touchée? Valeur true si le lutin touche la couleur.	souris pressée?

- Toutes les **variables** prédéfinies qui sont modifiables par **script** ne le sont qu'à l'aide de **blocs de commande** spécifiques et de même couleur que la **variable** à modifier.
Exemple : Le **bloc** 

permet de modifier la **variable** 

.

Nouvelles variables

- En plus des **variables** prédéfinies, on peut à partir de l'onglet Données créer ses propres **variables** et des **listes** de variables. Une **variable** créée peuvent être liée au **lutin** associé à la **fenêtre de scripts** où elle est créée, elle correspond alors à une nouvelle propriété du **lutin** et ne peut pas être utilisée ou modifiée dans les autres **fenêtres de scripts**.
Sauf dans des cas très particuliers, à la création d'une **variable** il est préférable de choisir l'option ◉ Pour tous les lutins , la **variable** est alors utilisable par tous les **scripts**.

- Il existe deux types de **blocs de commande**, 

et 

,qui permettent de modifier la valeur des **variables** créées en sélectionnant leur noms.
- A sa création, une **variable** ou une **liste** peut être montrée ou non sur la **scène**, il suffit pour cela de cocher ou non la case qui précède son nom. Exemple : 

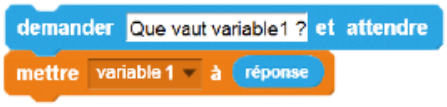
.
Des **blocs de commande** permettent aussi de montrer ou de cacher une **variable** ou une **liste**. Exemples : 

et 

.
- Pour initialiser une **variable** avec une valeur entrée par l'utilisateur du programme, il faut utiliser le **bloc** 

de l'onglet Capteurs , le programme attend que l'utilisateur entre une valeur (texte ou nombre) puis cette valeur sera affectée à la **variable** 

qui est prédéfinie. On peut ensuite affecter la réponse à une autre **variable** préalablement créée en assemblant deux **blocs** comme ci-dessous :



- A une **variable** non prédéfinie ont peut affecter un nombre, une chaîne de caractère (texte) ou un booléen (**true** ou **false**) et la modifier en utilisant l'un des deux **blocs de commande** suivants : 

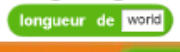
ou 

Les blocs opérateurs

- Pour faire des opérations avec des variables numériques il faut utiliser les **blocs opérateurs** suivants disponibles sur l'onglet Opérateurs :

Le dernier **bloc opérateur** permet d'utiliser les fonctions mathématiques usuelles.
On remplit les champs de ces **blocs opérateurs** en y faisant glisser des **variables** prédéfinies ou les **variables** créées depuis l'onglet Données ou en y saisissant directement des nombres.
Les **blocs opérateurs** peuvent être imbriqués les uns dans les autres .

- Il existe trois **blocs opérateurs** spécifiques pour l'utilisation de chaînes de caractères :


: mise bout à bout , 

: nombre de caractères , 

Exemple d'utilisation : 

- Il existe six **blocs opérateurs** qui renvoient des booléens (valeur **true** ou **false**):


et

Ces **blocs opérateurs** de forme hexagonale sont utilisés dans les **blocs** d'instructions conditionnelles.Certains peuvent s'imbriquer dans les autres : 

Les boucles et instructions conditionnelles.

- Il existe cinq blocs prévus pour recevoir d'autres piles de blocs de commandes à exécuter en boucle ou sous une condition donnée. On les trouve dans l'onglet Contrôle.

Les trois derniers **blocs** ont un emplacement hexagonal destiné à recevoir un **bloc opérateur** booléen.

Les objets listes

- A la création d'une **liste** de variables, les **blocs** de ses **variables** et de ses méthodes utilisables apparaissent dans l'onglet Données :

  Variables utilisables: 

  Méthodes utilisables :

- Les éléments d'une liste peuvent être des variables de types différents, il est même possible de créer des listes qui contiennent des listes.

Les clones

- Il est possible de créer des **clones** d'un **lutin** en utilisant dans un **script** d'un objet (scène ou lutin) autre que le **lutin** à cloner le **bloc** 

.

  On peut alors créer des **scripts** dans la **fenêtre des scripts** du **lutin** cloné en utilisant le bloc 

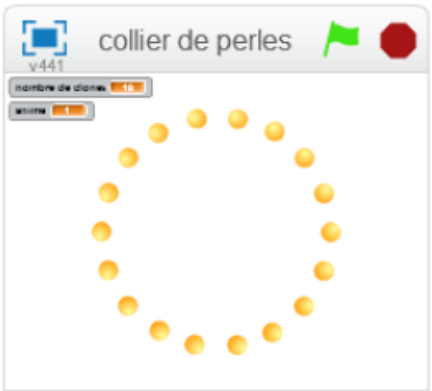
pour ajouter des méthodes à ses **clones** ou changer ses propriétés sachant que tous les **scripts** du **lutin** cloné s'appliquent aussi à tous ses **clones**.

- Attention si on crée *n* fois de suite un **clone** d'un **lutin** en utilisant le bloc 

dans la propre **fenêtre de scripts** du **lutin** à cloner, on obtient au final 2^n **clones**. En effet dans ce cas, à chaque création d'un nouveau **clone**, le **lutin** cloné accompagné de son **clone** se substitue au **lutin** cloné, ce qui peut très rapidement saturer la mémoire de l'ordinateur.
- Il peut être judicieux de prévoir des **scripts** pour détruire les **clones** qui ne sont plus utiles en utilisant le bloc 

par exemple lorsqu'un **clone** sort de la **scène**.
- On ne peut pas créer plus de 301 **clones**.
- Exemple d'un programme utilisant des **clones** :

La scène avant de cliquer sur le drapeau vert



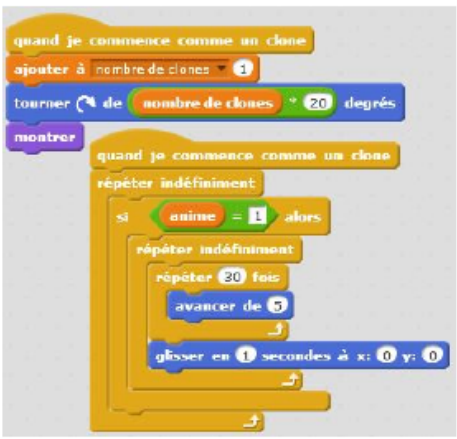
Après un clic sur le drapeau vert, le cercle des 18 clones de la balle se dilate et se contracte en continu.

Ce **script** crée 18 clones du lutin Ball.

Script de la scène

Le premier script actualise le nombre de clones créés et fixe la direction du clone créé.

Le deuxième script attend que les 18 clones soient tous créés avant d'animer le clone créé : alternance de dilatations de 30×5 unités et de contractions jusqu'à l'empilement des 18 clones au centre de la scène.



Les deux scripts du lutin Ball

Dessiner sur la scène avec un lutin.

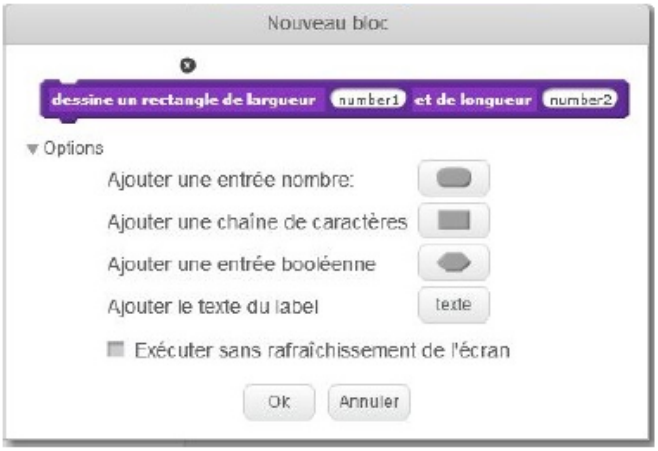
- On peut avec chaque **lutin** (*lorsque l'un de ses scripts change ses coordonnées*) dessiner un segment sur la **scène** en utilisant son **stylo** (*s'il est en position d'écriture*), l'origine du segment est la position initiale du **lutin** et l'extrémité sa position finale ; la pointe du **stylo** étant le **centre** du **lutin**. On peut changer la couleur la taille et l'intensité du **stylo** à l'aide des blocs de commande de l'onglet Stylo. Ci-dessous quelques exemples de blocs utilisables :

  Pour changer la couleur dans le dernier bloc, il suffit de cliquer sur la case de couleur puis de cliquer sur un endroit de l'écran où la nouvelle couleur apparaît.
- A l'aide du **bloc** estampiller on peut aussi coller sur la **scène** une image du **lutin** à l'endroit où il se trouve en gardant les dimensions et l'orientation actuelles du **lutin**.

Création de nouveaux blocs de commande.

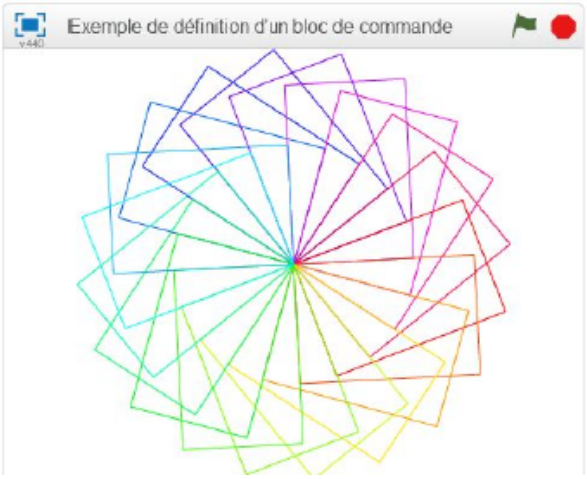
- Dans la **fenêtre des scripts** d'un **lutin** on peut créer de nouvelles méthodes pour ce **lutin** et leurs **blocs de commande** correspondants. Ces **blocs** qui peuvent recevoir des paramètres ne seront utilisables que dans la fenêtre des scripts où ils sont créés.
- Exemple d'utilisation d'un **nouveau bloc de commande** et du **stylo**:

Boite de dialogue à la création du nouveau **bloc de commande**.



Dans la **fenêtre des scripts** d'un **lutin**, figure la définition du nouveau **bloc** et un **script** qui utilise ce nouveau **bloc**.

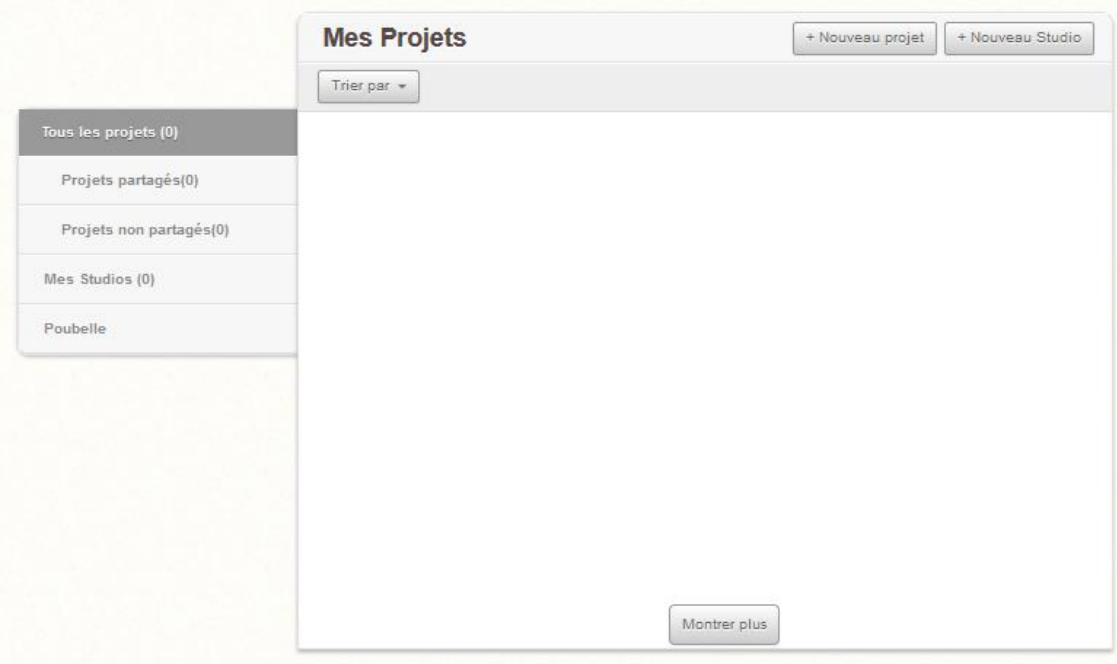
Voici le dessin obtenu sur la **scène** après avoir cliqué sur le drapeau vert.



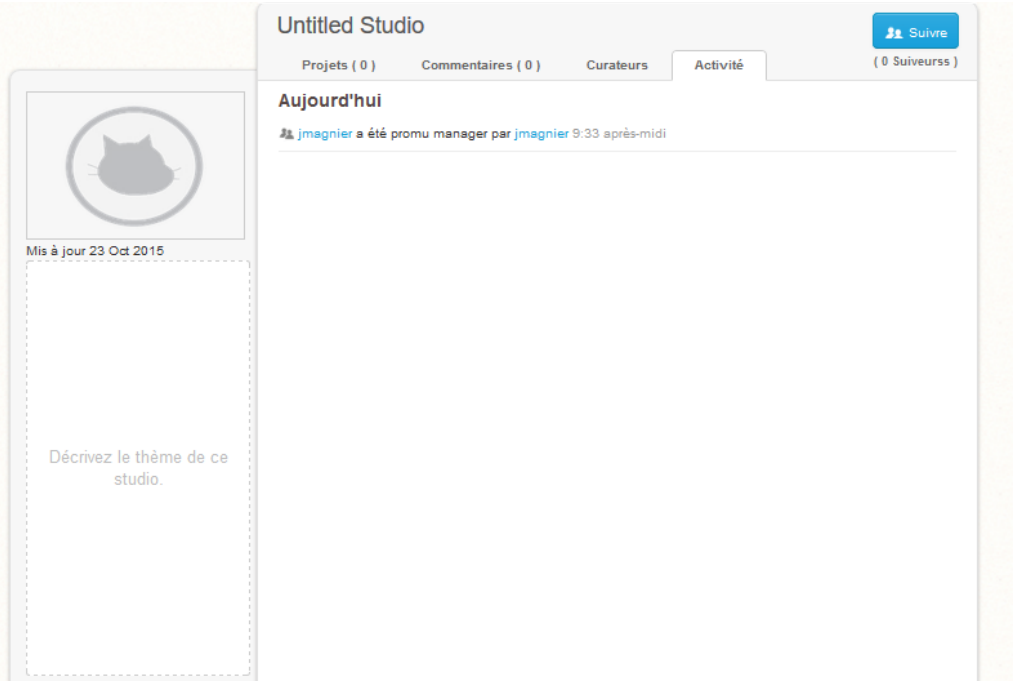
Créer un **STUDIO** pour ma classe

A quoi sert un studio ?
Les studios Scratch sont un moyen de rassembler et d'organiser les projets Scratch en ligne, et de partager les projets des élèves.

1. Se rendre sur le site de SCRATCH et se connecter avec ses identifiant et mot de passe.
2. Cliquer sur l'icône  . Voici la page obtenue :



3. Cliquer sur l'icône 
4. Paramétrer le studio (donner un titre, une description rapide du studio) et ajouter des projets (les votres et ceux des élèves ...)



Pour ajouter des élèves au STUDIO, il faut les inviter en tant que "**curateurs".**

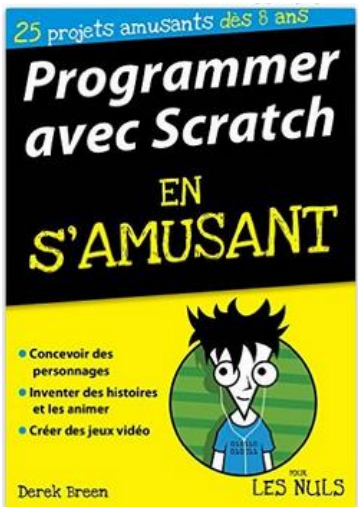
Les élèves doivent disposer d'une adresse mail valide.

Livres

Scratch pour les kids : initiation à l'informatique pour les 8-12 ans
Dès 8 ans - The LEAD Project (Auteur)
Paru le 5 mars 2015 - broché- 15,90€

Cahier d'activités, scratch
F. Pain (Auteur)
Paru le 18 juin 2015 - broché- 12€

Python pour les kids la programmation accessible à tous
La programmation accessible aux enfants ! Dès 10 ans - J.R. Briggs (Auteur) - Paru en mars 2015 - Guide(broché) 22,90€



Programmer avec Scratch en s'amusant, mégapoche pour les Nuls Broché – 8 octobre 2015
de Derek BREEN
Editeur : First Interactive (8 octobre 2015) 17,95€

Un ensemble de ressources autour de SCRATCH, à cette adresse

http://fr.padlet.com/jmagnier/scratch