



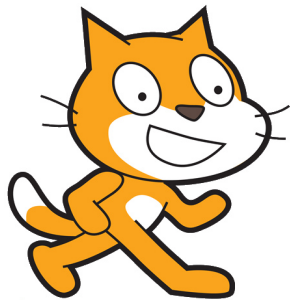
Introduzione alla programmazione

SCRATCH

Hackspace

Festival letteratura

Mantova



Stefano Lacaprrara

INFN Padova



• Io: Stefano Lacaprara:

- [stefano.lacaprara@pd.infn.it](mailto:stefano.lacaprara@pd.infn.it) 3493723643

• Web page:

- <http://www.pd.infn.it/~lacaprar/ProgettoScuola/Scratch/>

- Contiene links alla pagina web di scratch,

- Documentazione varia,

- Le slide e relativi progetti scratch saranno caricati dopo ogni incontro

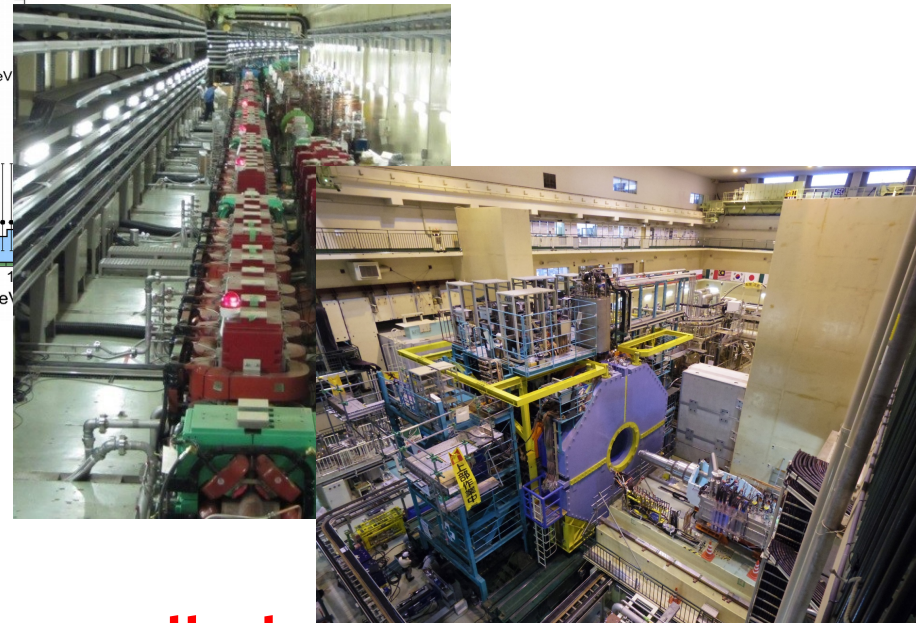
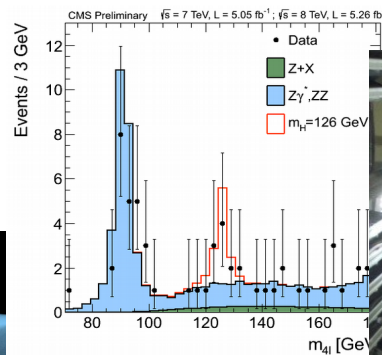
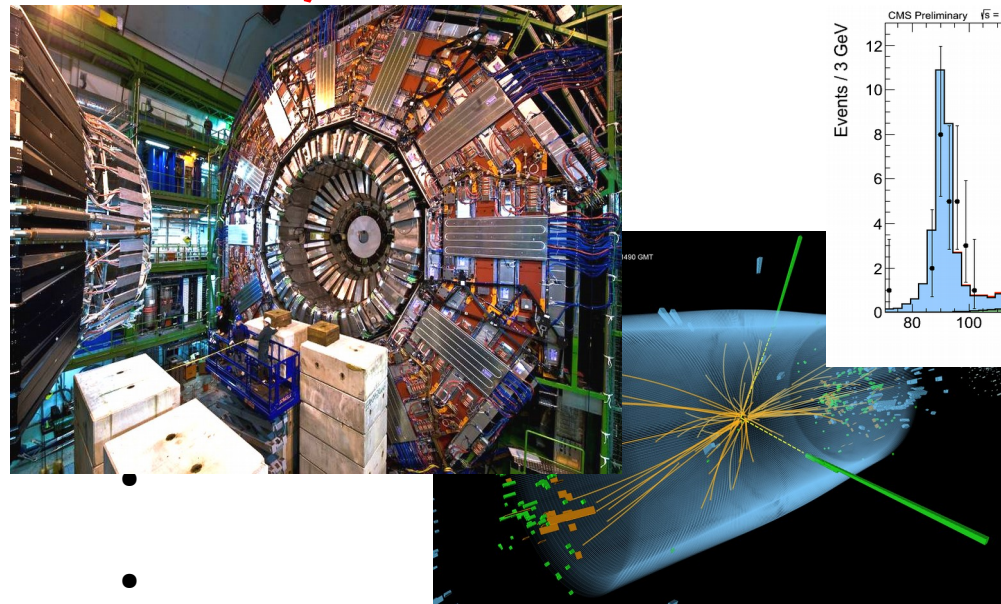
- Bibliografia

- ....

# Mi presento



- Sono un ricercatore dell'Istituto Nazionale di Fisica Nucleare (INFN)
- Lavoro nel campo delle particelle elementari, nella collaborazione CMS, al CERN di Ginevra e BelleII a KEK Tsukuba (Giappone)



- Oltre al lavoro di fisico, e strumentale a quello, ho sviluppato per molti anni software a vari livelli sia direttamente per la ricostruzione della fisica degli apparati sia per il calcolo distribuito
- Ho insegnato e insegno Meccanica, termodinamica, elettromagnetismo e programmazione (C++) all'universita' di

# Perche' questo corso



- **Viviamo in un mondo scientifico/tecnologico, ma l'interesse per la scienza/tecnologia non e' diffuso come potrebbe e dovrebbe.**
  - Questo e' un problema per la scienza e quindi per il mio campo specifico,
  - Ma e' un problema anche per il nostro futuro
- **Perche' succede?**
  - Ci sono tante cause, una di queste e' la mancanza di stimoli adeguati nell'eta' dell'apprendimento
- **La presenza di scienza e tecnologia a scuola e' purtroppo limitata, non ultimo per questioni di risorse**
  - Spesso e' presentata con materiali poveri e in ambiti non immediatamente collegati alla vita tecnologica che i bambini vivono.



- **Esempi di esperimenti di scienza con bicchieri di carta, carrellini, fagioli, etc, quando i bambini usano tutti i giorni computer, smartphone, telecomandi, ...**
  - **Si crea una distanza tra quello che un bambino percepisce come scienza e quello che la scienza e' veramente.**
  - **Da questa distanza, l'iniziale interesse e curiosita' che hanno tutti i bambini verso tutto, e quindi anche verso scienza e tecnologia, puo' calare.**
    - **Per diventare indifferenza o, peggio, avversione.**

# Prenderli da piccoli



- **Non e' solo un cliché', ma una vera necessita'.**
- **Diversi studi mostrano che un bambino "decide" molto presto se ha interesse per scienza e tecnologia (e matematica e tutto quello che gli sta attorno)**
- **L'attitudine verso la scienza e' un atteggiamento che il bambino acquisisce (o non acquisisce) presto, tipicamente tra i 8 e gli 11 anni, e tipicamente non lo cambiera' piu'.**
  - **Se decide che non gli interessa, decidera' anche inconsciamente che la scienza, matematica, informatica, etc e' troppo difficile, e non la imparera' in futuro.**
  - **C'e' inoltre una grande differenza di genere, tra maschi e femmine, che e' probabilmente la singola variabile piu' importante in relazione all'attitudine verso la scienza**
  - **Oltre elementi culturali, sociali, familiari etc, etc...**



- **Se c'e' un singolo campo onnipresente nella nostra vita oggi e' l'informatica.**
  - Nel bene (e nel male!) il software e' una parte grandissima della nostra vita
  - E' una parte fondamentale della vita di qualunque scienziato
- **Programmare e' una attivita' creativa!**
  - C'e' uno stereotipo imperante che associa un programmatore ad un nerd, o disadattato sociale o comunque ad una figura piuttosto negativa.
- **Programmare vuol dire analizzare e risolvere problemi:**
  - Per risolverli, occorre pensare, decidere, e valutare i nostri sforzi.

# Nativi digitali



- **Spesso si parla dei bambini di oggi come nativi digitali, perché sanno usare iCosì o computers prima e a volte meglio di noi.**
  - **E' una definizione assolutamente infelice, perché da' l'impressione che i bambini controllino il mondo digitale. Non e' così'.**
- **Certo, sanno usare videogiochi o smartphone già a 3 anni: ma si chiedono cosa c'e' dentro un computer?**
  - **Si rendono conto che un computer (o cellulare o telecomando o ...) e' stato costruito e programmato da qualcuno?**
  - **Hanno mai provato a programmare un computer?**
  - **O si limitano ad usare passivamente cose preparate da altri?**



# Obiettivi progetto scratch



- **L'obiettivo primario di questo progetto è dare ai bambini la consapevolezza che è possibile interagire con gli strumenti tecnico informatici che li circondano non solo in maniera passiva, ovvero semplicemente usandoli, ma anche in maniera attiva, cioè programmandoli direttamente.**
- **L'attività vuole essere presentata come un gioco, da fare singolarmente e collettivamente, produrre un risultato tangibile in tempi brevi e allo stesso tempo suscitare la curiosità del bambino ai temi dell'informatica, della programmazione, e, più in generale, ad interessarsi a cosa c'è dentro, in questo caso a livello software, i vari strumenti tecnologici che lo circondano.**
- **L'analogo hardware può essere coperto da altri progetti orientati all'elettronica, meccanica.**

# Schema del workshop



- **La mia idea e' di introdurre la piattaforma di programmazione SCRATCH a chi segue il workshop**
  - **Mostrare l'interfaccia di SCRATCH e come si usa**
  - **Usando come esempio concreto un semplice programma da fare insieme**
    - **Ogni persona che segue il workshop dovrebbe riuscire, con opportuno aiuto e guida, ad avere un programma funzionante**
    - **L'istruttore spiega cosa vuole fare (obiettivi), come dividere azioni complesse in pezzi semplici (analisi), e infine un possibile modo di farlo (coding)**
      - **Deve essere chiaro che non e' l'unica soluzione possibile: eventuali soluzioni o proposte alternative vanno discusse e incoraggiate**
      - **Allo stesso modo modifiche ed estensioni dell'obiettivo iniziale sono molto ben accolte.**

# Schema del workshop



- **Dopo il primo esercizio insieme, che dovrebbe permettere a tutti di capire come funziona scratch e introdurli alla programmazione, si possono proporre sfide/esercizi piu' o meno complessi a seconda dell'audience (eta'/preparazione/interesse/etc).**
  - **In caso di audience diversificata si possono proporre diverse sfide**
    - **Ne ho preparate un buon numero, via via piu' complesse**
      - Per alcune e' necessario introdurre elementi di programmazione (es interazione tra sprite, messaggistica, variabili, etc) che non ci sono nel primo esempio fatto insieme
      - In questo caso e' probabilmente necessario spiegare questi elementi.

# Schema del workshop



- **Non sia una lezione frontale!**
  - Tutto questo immagino sia molto ovvio
  - L'insegnante spiega alla lavagna/proiettore e gli studenti prendono appunti, e/o copiano quello che fa l'insegnante replicando il suo programma
    - Così si impara poco, e non è un workshop
- **Meglio proporre l'esercizio, spiegando l'obiettivo e poi:**
  - fare l'analisi, possibilmente insieme, e coinvolgendo l'audience.
    - Se durante l'analisi viene fuori la necessità di nuovi elementi, introdurli con un esempio
  - Una volta fatta l'analisi, concentrarsi su ciascun pezzo dell'algoritmo, partendo dai più semplici
    - I partecipanti a questo punto dovranno iniziare a programmare, implementando questi pezzi del programma

# Schema del workshop



- **Diamo il tempo necessario per riuscire a completare almeno in parte il programma**
  - **E' piu' importante che ciascuno riesca a fare un programma anche banale da solo che riesca a copiarne uno complesso!**
  - **Via via che le persone riescono/non riescono ad andare avanti, proponiamo una soluzione al proiettore**
    - **Discutiamo soluzioni alternative**
-

# Cosa e' SCRATCH



- **Impara a programmare mentre crei storie interattive, giochi, progetti multimediali.**
- ***Start from scratch* vuol dire *partire da zero***
  - **E' una delle idee di base di scratch: non serve saper programmare**
- **La programmazione e' completamente grafica: non occorre sapere nulla di programmazione formale, ma si deve solo costruire strutture semplici (o piu' complesse) mettendo insieme mattoncini colorati.**
  - **E' pensata come un gioco, ma richiede che il bambino pensi a cosa vuole fare e sia in grado di dividere azioni complesse in pezzi elementari.**
  - **E' fortemente multimediale**

# Cosa si puo' fare



- **Qualche esempio di progetti con SCRATCH**
  - **Personaggi che si muovono e interagiscono su un palcoscenico,**
  - **I personaggi interagiscono con l'utente:**
    - **Sei tu che comandi il computer**
  - **Interazioni tra personaggi**
  - **Slideshow interattivo,**
  - **Storie interattive**
  - **Simulazioni**
  - **Videogames,**
  - **...**
- **Ci sono infinite applicazioni che fanno queste cose, ma il punto fondamentale e' che sara' il bambino a programmare il computer, a comandare il computer per fargli fare quello che vuole.**



- **Un PC: non serve particolarmente nuovo**
  - **Monitor, mouse, tastiera, etc**
  - **Sistema operativo:**
    - **Windows, ChromeOS, Mac, or Linux**
    - **Adobe Flash Player recente (rilasciata dopo giugno 2016)**
    - **Web-browser recente (Chrome, Firefox, Safari, Edge, IE11)**
  - **CPU e RAM: se riesci a far funzionare il S.O., allora tipicamente il PC e' sufficiente. Computer molto vecchi (e lenti) potrebbero eseguire SCRATCH lentamente.**
  - **Suono/Video: altoparlanti o cuffie e microfono (per registrare), webcam per video/foto. Anche via USB**
  - **rete**



# Installazione



- **La versione attuale di SCRATCH e' la 2.0**
  - **Scratch 2.0 e' pensato per funzionare online, dal browser.**
    - **Richiede quindi una connessione di rete decente**
    - **La webapplication e' basata su flash (!), quindi occorre che sia installato sul client**
    - **Il che implica che non puo' funzionare da smartphone/tablet iOS/Android (no flash)**
  - E' anche possibile installarlo in modalita' offline
  - In alternativa e' possibile usare la versione precedente 1.4, che e' nativamente offlin
- Per usare la versione online, basta andare su:
  - <https://scratch.mit.edu/> : Try it out

# Installazione offline



- <https://scratch.mit.edu/download>
- Servono: Adobe AIR + Scratch offline editor (+ support material)
  - Ci sono versioni per win, MacOS, linux (32 bit)
    - NB: AdobeAir non e' piu' supportato per linux (!), ma si puo' usare l'ultima versione disponibile (2.6)
- **NB: se linux e' 64 bit, l'installazione e' un po' piu' complessa**

# Installazione offline 64 bit



## Io ho seguito le istruzioni qui

- <https://askubuntu.com/questions/913892/how-to-install-scratch-2-on-ubuntu-16-10-or-17-04-64bit/913912>

```
# install necessary i386 libraries
$ sudo apt-get install libgtk2.0-0:i386 libstdc++6:i386 libxml2:i386 libxslt1.1:i386 libcanberra-gtk-module:i386 gtk2-engines-
murrine:i386 libqt4-qt3support:i386 libgnome-keyring0:i386 libnss-mdns:i386 libnss3:i386

# make keyring visible for Adobe Air
$ sudo ln -s /usr/lib/i386-linux-gnu/libgnome-keyring.so.0 /usr/lib/libgnome-keyring.so.0
$ sudo ln -s /usr/lib/i386-linux-gnu/libgnome-keyring.so.0.2.0 /usr/lib/libgnome-keyring.so.0.2.0

# Download Adobe Air
cd ~/Downloads
wget http://airdownload.adobe.com/air/lin/download/2.6/AdobeAIRSDK.tbz2
$ sudo mkdir /opt/adobe-air-sdk
$ sudo tar jxf AdobeAIRSDK.tbz2 -C /opt/adobe-air-sdk

# Download Air runtime/SDK from Archlinux
wget https://aur.archlinux.org/cgit/aur.git/snapshot/adobe-air.tar.gz
$ sudo tar xvf adobe-air.tar.gz -C /opt/adobe-air-sdk
$ sudo chmod +x /opt/adobe-air-sdk/adobe-air/adobe-air

# Get actual scratch file URL from https://scratch.mit.edu/scratch2download/
$ sudo mkdir /opt/adobe-air-sdk/scratch
wget https://scratch.mit.edu/scratchr2/static/sa/Scratch-456.0.4.air
$ sudo cp Scratch-456.0.4.air /opt/adobe-air-sdk/scratch/
cp Scratch-456.0.4.air /tmp/
cd /tmp/
unzip /tmp/Scratch-456.0.4.air
$ sudo cp /tmp/icons/AppIcon128.png /opt/adobe-air-sdk/scratch/scratch.png
```

# Installazione offline 64 bit (2)



## Creo launcher:

```
sudo vim /usr/share/applications/Scratch2.desktop

[Desktop Entry]
Encoding=UTF-8
Version=1.0
Type=Application
Exec=/opt/adobe-air-sdk/adobe-air/adobe-air /opt/adobe-air-sdk/scratch/Scratch-456.0.4.air
Icon=/opt/adobe-air-sdk/scratch/scratch.png
Terminal=false
Name=Scratch 2
Comment=Programming system and content development tool
Categories=Application;Education;Development;ComputerScience;
MimeType=application/x-scratch-project
```

- **Alla fine posso lanciare scratch da panel usando il launcher.**
- 
- **Alternativa molto piu' semplice: uso versione web o installo scratch 1.4**

# Installazione scratch 1.4



- [https://scratch.mit.edu/scratch\\_1.4/](https://scratch.mit.edu/scratch_1.4/)
  - **Puo' essere utile per pc particolarmente vecchi o se il 2.0 non funziona bene**
  - **1.4 e 2.0 sono compatibili**
  - **Per windows:**
    - **Scaricare <http://download.scratch.mit.edu/ScratchInstaller1.4.exe> (33MB)**
    - **Doppio click sul .exe e seguire le istruzioni:**
      - **Windows si lamentera' che l'eseguibile non e' formato e potrebbe essere pericoloso. Cliccare su "ok – installare lo stesso – mi fido di questa sorgente"**
      - **Dobbiamo poter scrivere nella cartella dove stiamo estraendo scratch: se ci sono errori (error opening file for writing), provare ad estrarre scratch in desktop**
    - **L'eseguibile si trova dentro la cartella scratch (e nel menu start)**
  - **Per Mac/OS**
    - **<http://download.scratch.mit.edu/MacScratch1.4.dmg>**
    - **Spostare scratch dentro cartella applicazioni**
  - **Per Ubuntu**
    - **Da Software Center: cerca SCRATCH e installa (servono diritti di amministratore)**
    - **Oppure si scarica pacchetto .deb (e relative dipendenze)**

# Pagine web



- **Pagina principale:**  
<http://scratch.mit.edu/>
  - Scaricare il programma,
  - documentazione,
  - Forum degli utenti (c'e' anche sezione in italiano, non frequentatissima)
  - migliaia di progetti caricati dagli utenti
    - I progetti si possono eseguire sul web (richiede flash)
    - O scaricare ed eseguire e modificare localmente

FF ShFF STL ROOT Fun Python SL home

**SCRATCH** home projects galleries support forums about Language

Imagine • program • share Login or Signup for an account search

Create and share your own interactive stories, games, music, and art

Check out the 3,078,505 projects from around the world!

To create your own projects:  
Download Scratch

Featured Projects [See more](#)

Scratch 2.0 Beta  
Try It! Try out the Scratch 2.0 Beta, and help us make it better!  
[Check it out](#)

ScratchEd  
Do you help people learn Scratch? Join ScratchEd, our new online community for educators.  
[Find out more](#)

Video Tutorials  
Check out our new collection of intro video tutorials.  
[Learn more](#)

Projects Selected by joletole [Learn more](#)

# Pagine web



- <http://scratched.media.mit.edu/>
  - Pagina dedicata agli educatori. Gran parte in inglese
  - Storie, news
  - Risorse (documentazione, tutorial, corsi, ...)
  - Discussioni (forum)
  - Eventi (Workshop, Meetups, Webinars, ScratchDay)

The screenshot shows the ScratchEd website interface. At the top, there's a navigation bar with 'Stories', 'Resources', 'Discussions', 'Members', and 'Events'. Below this, there are two main sections: 'What is Scratch?' and 'What is ScratchEd?'. The 'What is Scratch?' section includes a brief description and a 'Learn more' link. The 'What is ScratchEd?' section describes it as an online community for educators and includes icons for 'share stories', 'exchange resources', 'ask questions', and 'find people'. To the right, there's a video player for 'Get Started with Scratch'. Below these sections, there are three columns: 'New Stories', 'New Resources', and 'New Discussions', each containing a list of recent items with thumbnails and brief descriptions.



Getting Started with  
**SCRATCH**  
version 1.4



<http://scratch.mit.edu>



# La schermata di SCRATCH



Scratch 2 Offline Editor

ScrATCH File Modifica Suggestimenti Info

v466.0.4

Script Costumi Suoni

**Movimento** Situazioni  
Aspetto Controllo  
Suono Sensori  
Penna Operatori  
Variabili e Liste Altri Blocchi

fai 10 passi  
ruota di 15 gradi  
ruota di 15 gradi  
punta in direzione 90  
punta verso puntatore del mouse  
vai a x: 0 y: 0  
raggiungi puntatore del mouse  
scivola in 1 secondi a x: 0 y:  
cambia x di 10  
vai dove x è 0  
cambia y di 10  
vai dove y è 0  
rimbalza quando tocchi il bordo

x: 0  
y: 0

Sprite Nuovo sprite:

Stage 1 sfondo

Nuovo sfondo:

Sprite1

x: 240 y: -77

Scratch 2 Offline Editor interface showing a stage with a Scratch cat sprite, a script area with movement blocks, and a sprite area with a Scratch cat sprite.

# La schermata di SCRATCH



Scratch 2 Offline Editor

SCRATCH File Modifica Suggerimenti Info

Script Costumi Suoni

Movimento Situazioni  
Aspetto Controllo  
Suono Sensori  
Operatori

x: 0  
y: 0

Sprite

Nuovo sp

Stage  
1 sfondo

Nuovo sfondo:

Sprite1

rimbalza quando tocchi il bordo

Menu principale:

- Scegli lingua (default e' quella di sistema)
- Menu file
  - Nuovo progetto, apri, salva, importa/esporta
  - Condividi (per caricare sul sito di scratch)
- Modifica
- Suggerimenti
  - Mostra tutorial per diversi progetti
- Info (apre pagina web)

# La schermata di SCRATCH



Scratch 2 Offline Editor

File Modifica Suggestimenti Info

Script Costumi Suoni

Movimento Situazioni  
Aspetto Controllo  
Suono Sensori  
Penna Operatori  
Variabili e Liste Altri Blocchi

fai 10  
ruot  
ru  
pu  
pu  
va  
ra  
sc  
ca  
va  
ca  
va  
rimbal

Sprite1

Stage 1 sfondo

Nuovo sfondo:

x: 240 y: -77

x: 0  
y: 0

**Palcoscenico (stage)**

- Dove andremo a vedere i risultati dei nostri programmi
- Gli attori sono gli “sprite” (figurine?)
  - Per esempio questo gatto
- I tasti verde e rosso in alto a destra serviranno per far partire e fermare la scena
- Si puo' mettere a schermo intero
  - Notare in basso a dx le coordinate x-y dello sprite.

# La schermata di SCRATCH



Scratch 2 Offline Editor

File Modifica Suggestimenti Info

Scratch v466.0.4

Script Costumi Suoni

**Movimento** Situazioni  
Aspetto Controllo  
Suono Sensori  
Penna Operatori  
Variabili

**Gli sprite**

- E' l'elenco di tutti gli sprite che ho creato
- In questo caso, solo uno
- Posso aggiungerne altri

In piu' c'e' lo stage (sfondo), che non e' uno sprite, ma puo' essere comandato: le scene

Sprite Nuovo sprite: [icone]

Stage 1 sfondo

Nuovo sfondo: [icone]

Sprite1

X: 240 Y: -77

rimbalza quando tocchi il bordo

# La schermata di SCRATCH



**Libreria delle azioni**

- Ci sono le azioni che gli sprite possono compiere
  - Sono i mattoncini elementari dei nostri programmi.
  - Un programma complesso sarà una costruzione di mattoncini elementari
- Sono divise per tipologia
  - Movimento, aspetto, suono, etc
  - Ogni categoria ha diversi elementi
  - Ogni elemento ha una forma particolare che si può incastrare o meno con altri elementi
  - Alcuni elementi hanno al loro interno “spazio” per parametri oppure per altri elementi
    - Sarà più chiaro usandoli

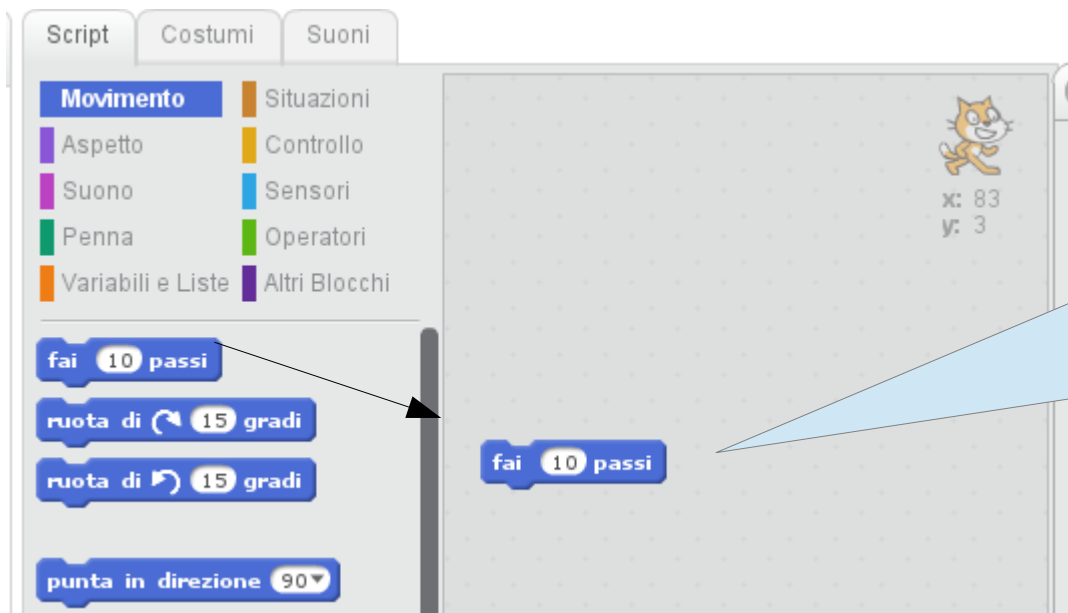
# La schermata di SCRATCH



**Area di programmazione**

- Qui andremo a creare il nostro programma
- Per farlo, si prendono i mattoncini dalle librerie, e si trascinano nell'area di script
  - Se le forme sono compatibili, i mattoncini si uniranno quando li trasciniamo vicini
  - I colori dei mattoncini identificano le tipologie (movimento, aspetto, suono, etc)
- Da notare le tre linguette
  - Script: area delle azioni
  - Costumi: disponibili per lo sprite
  - Suoni: i suoni disponibili
- Da notare [Sprite].
  - Ci sarà un'area di programmazione per ogni sprite e per lo stage

# Iniziamo a programmare

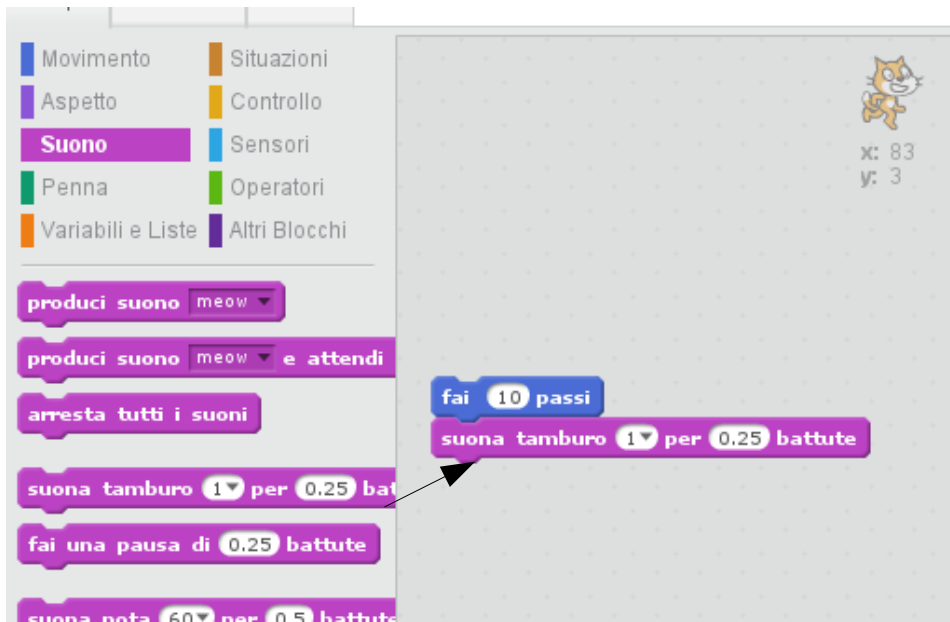


Trascina un mattoncino  
**fai 10 passi**  
nell'area di programmazione

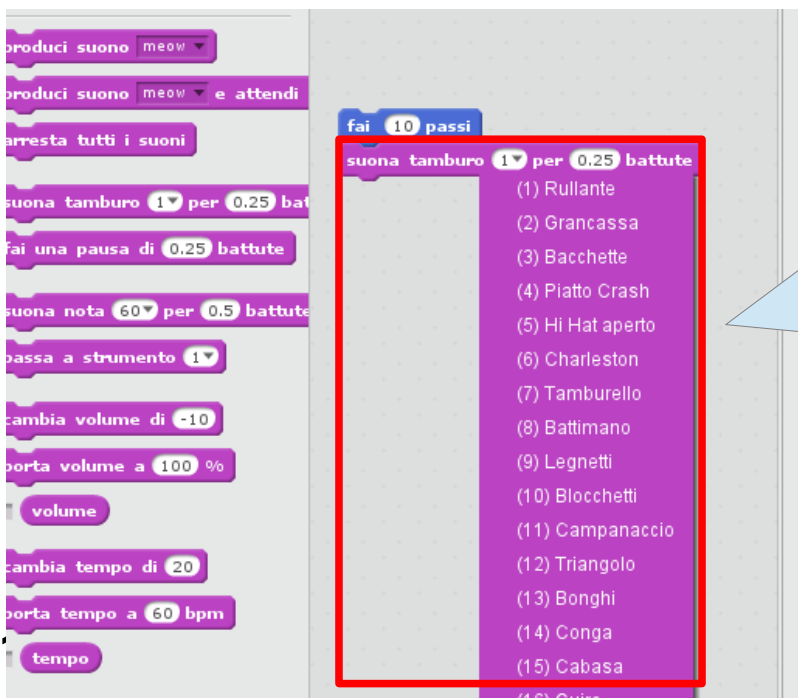


**Click** su mattoncino per far  
muovere il gatto di 10 passi

# Aggiungiamo un suono



- Dalla libreria **Suono** (viola) trascina
- **Suona tamburo 48 per 0.2 battiti**
- nell'area di programmazione appena sotto l'altro mattoncino.
- I mattoncini si incastrano perche' sono compatibili.
- Ho creato un programma con due azioni:
  - Fai 10 passi
  - Suona il tamburo
- **Click** su mattoncino per sentire il tamburo



- Alcune azioni (mattoncini) sono configurabili, cioe' posso cambiare alcuni parametri dell'azione stessa.
- Area bianca dentro il mattoncino:
  - Fai N passi: default e' 10, ma posso cambiarlo
  - Suona tamburo X. Se clicco su X(default e' 48), mi compare una lista di tutti gli strumenti che posso suonare.
  - Per Y battute: per quanto tempo suonarlo.



# Primi passo di danza



- Ho aggiunto:
  - **Fai [-10] passi**
    - -10 vuol dire tornare indietro
  - **Suona altro tamburo**
- Adesso il mio gatto fa un passo avanti, suona un tamburo, fa un passo indietro e suona un tamburo diverso.
- Click su qualunque mattoncino per far partire il programma.
- **Pero' lo fa una volta sola!**
  - Potrei ripetere i 4 mattoncini ma non e' cosi' che si programma.

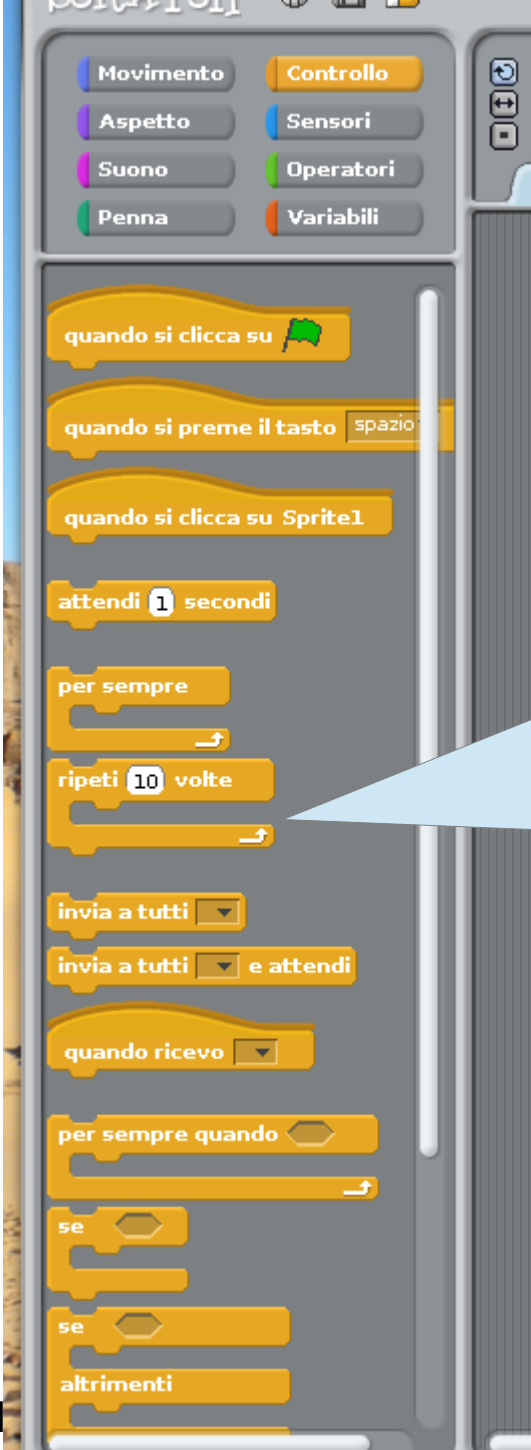
Introduco il concetto di **Iterazione (loop)**.

Ho un insieme di azioni che voglio ripetere N volte:

Raggruppo le azioni

Le inserisco dentro un modulo (mattoncino speciale) che ha come compito di ripetere una azione (o un insieme di azioni) N volte.

Indipendentemente dal tipo di azione



- Dalla libreria **Controllo** prendo
  - **Ripeti [10] volte**
- Attenzione alla forma del mattoncino:
  - Ha una tacchetta in alto, e una in basso, come i mattoncini di prima:
  - Si puo' incastrare prima o dopo gli altri mattoncini.
  - Ha spazio tra le due *ganasce* e le tacchette mostrano che puo' accogliere uno o piu' mattoncini.
  - Mentre lo trascino nell'area di programmazione, la forma cambia per poter accogliere i 4 mattoncini che ho gia' nell'area.



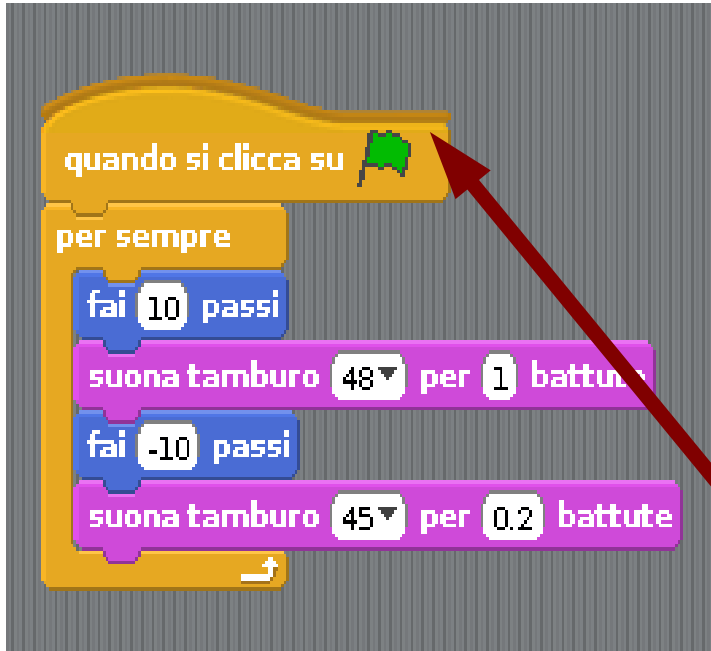
# Ripetere i passi all'infinito



- Dalla libreria **Controllo** prendo
  - **Per sempre**
  - E' un iteratore (ripete le azioni dentro le ganasce)
  - E le ripete per sempre (non per N volte)
  - Attenzione alla forma del mattoncino:
    - E' come **ripeti [10] volte** ma non ha la tacchetta in fondo
    - Visto che e' per sempre, non e' possibile fare un'altra azione dopo!
  - Per cambiare [ripeti 10 volte] con [per sempre]:
    - Trascino fuori i mattoncini dalle ganasce
    - Poi li metto dento l'altro loop
  - Per fermare [per sempre] click su bottone rosso in altri a destra



# Far partire il programma Bandiera Verde



- Per partire un programma in **[controllo]**, c'e' il mattoncino
- **Quando si clicca su [bandiera verde]**
- Attenzione alla forma del mattoncino:
  - Si possono attaccare mattoncini dopo, ma non prima.
- Ho un programma completo:
  - C'e' un bottone di partenza (e uno di arresto)
- **Proviamo a introdurre piu' interazione**

# Interazione



- Introduco un secondo programma
  - Indipendente dal primo (non c'e' collegamento tra i mattoncini)
- Viene attivato da un mattoncino controllo
  - **Quando si preme il tasto [spazio]**
    - Posso decidere di cambiare tasto
  - Azione e' cambiare **colore** di [25]
    - Ci sono diversi effetti grafici che si possono applicare su uno sprite
- Che cosa ci aspettiamo che succeda?

- Quando premo il tasto **verde**, il **programma 1 parte**, e lo sprite inizia a ballare.
- Se poi, in qualsiasi momento premo il **tasto spazio**, allora parte anche il **programma 2** (in parallelo) e lo sprite cambia colore.
- **Ho un programma che reagisce a stimoli esterni (tastiera): e' un programma interattivo!**

# Ripetere i passi all'infinito



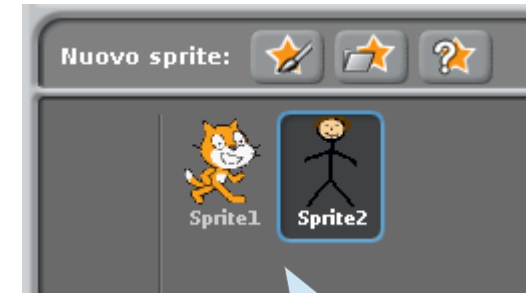
- Dalla libreria **Controllo** prendo
  - **Per sempre**
  - E' un iteratore (ripete le azioni dentro le ganasce)
  - E le ripete per sempre (non per N volte)
- Attenzione alla forma del mattoncino:
  - E' come **ripeti [10] volte** ma non ha la tacchetta in fondo
    - Visto che e' per sempre, non e' possibile fare un'altra azione dopo!
- Per cambiare [ripeti 10 volte] con [per sempre]:
  - Trascino fuori i mattoncini dalle ganasce
  - Poi li metto dento l'altro loop
- Per fermare [per sempre] click su bottone rosso in altri a destra



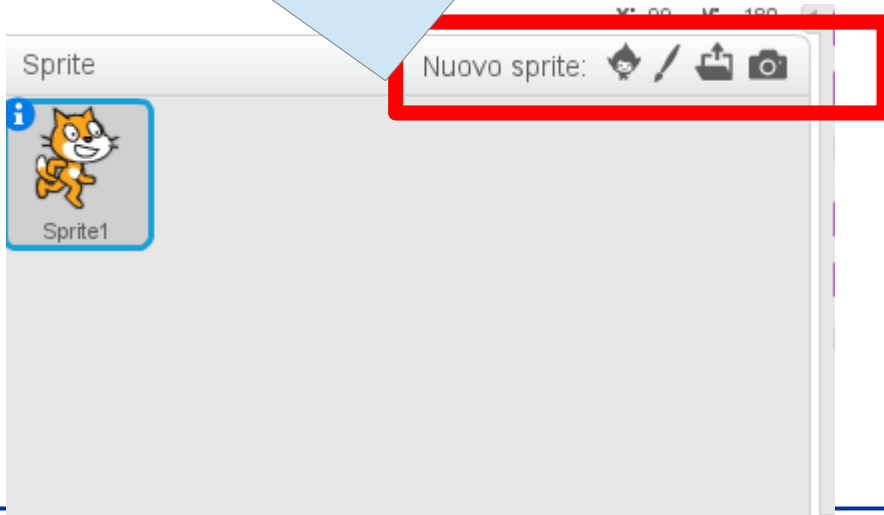
# Piu' sprite!



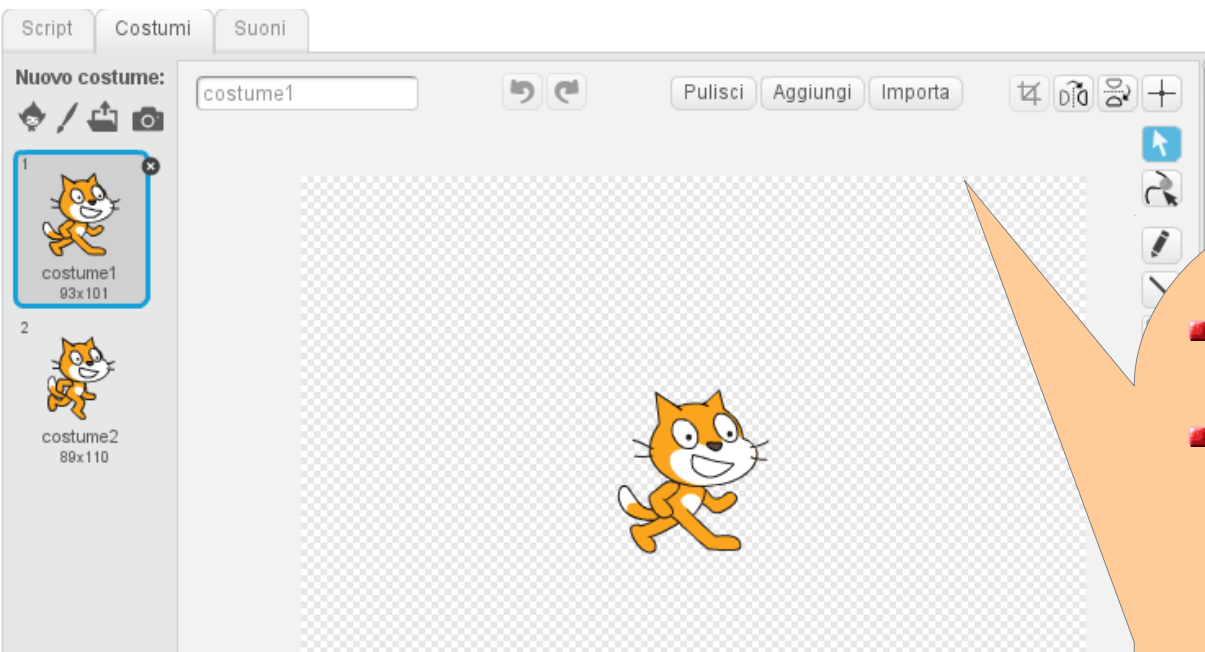
- Fino ad ora un solo sprite:
  - Posso usarne piu' di uno.
- Per aggiungerli, 4 possibilita':
  - **Carica dalla libreria**
  - **Disegna un nuovo sprite**
    - Si apre una finestra dove posso disegnarlo (simile a paint)
  - **Seleziona nuovo sprite da file**
    - Scratch ne ha molti disponibili, divisi per categoria (animali, fantasia, lettere, ...)
  - **Dalla webcam**
    - Fai una foto



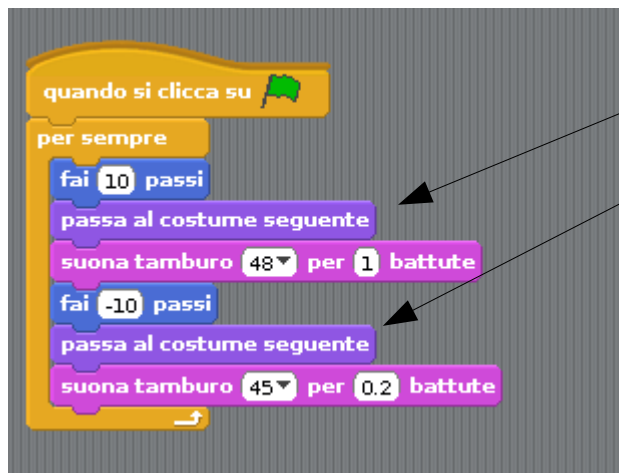
- **Adesso ho due sprite**
  - Il secondo l'ho disegnato io
- **Ogni sprite ha la sua area di programmazione:**
  - Il programma che abbiamo scritto prima appare solo se selezione lo sprite gattino (sprite1)
  - Il secondo sprite non ha ancora programmi associati.



# Animazione

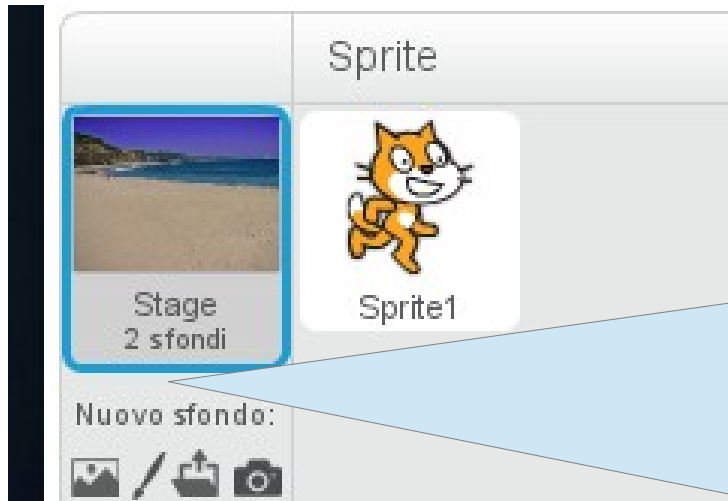


- Per animare uno sprite, si possono usare i **costumi**
- Ogni sprite puo' avere diversi costumi, che possono essere variazioni dello sprite originale
  - Per esempio figure vestite in modo diverso
  - Oppure figure in posizioni diverse
- Il gattino ha due costumi: alternandoli, possiamo animare lo sprite per farlo camminare.
  - Possiamo aggiungere altri costumi a piacere, disegnandoli da zero, o copiando uno gia' esistente e modificandolo.





# Lo sfondo (stage)



- Lo stage e' gestito come uno sprite speciale
  - Si possono definire script (programmi) per lo sfondo
    - Solo un insieme limitato di azioni sono possibili, per esempio, nessuna di movimento
  - Puo' avere diversi sfondi (analogo di costumi)
  - Si possono associare diversi suoni





- **Costumi o sfondi si possono:**

- **Disegnare**
- **Importare: una immagine gia' esistente (es foto)**
- **Acquisire con webcam collegata al pc (se presente)**



- Per creare uno sprite da webcam, occorre:
  - Prima disegnarne uno nuovo “finto”
  - Poi si seleziona lo sprite nuovo, e si sceglie [costumi]
  - Qui si crea un nuovo costume selezionando [web cam]
    - Volendo si modifica l'immagine acquisita
  - A questo punto si puo' cancellare il primo costume “finto”

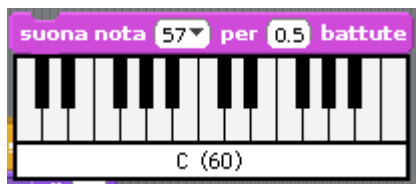




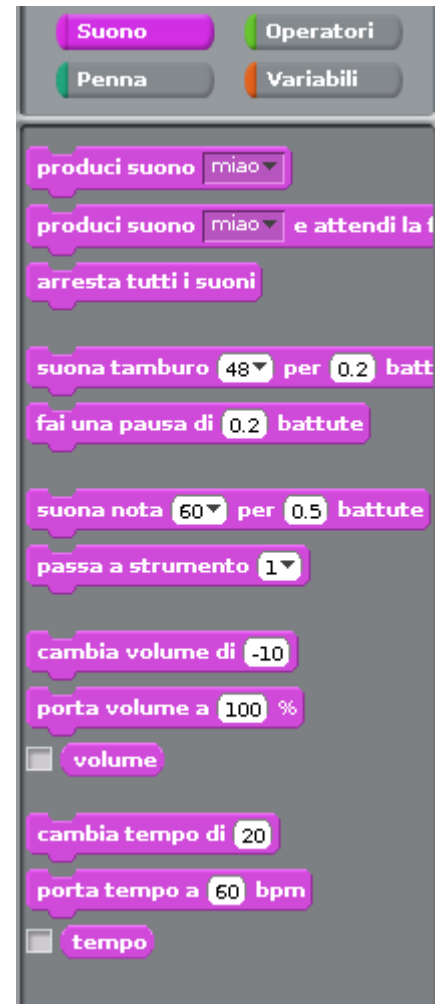
- **Ci sono diversi modi di far suonare scratch**

- **Uno l'abbiamo già visto: la libreria [suono]**

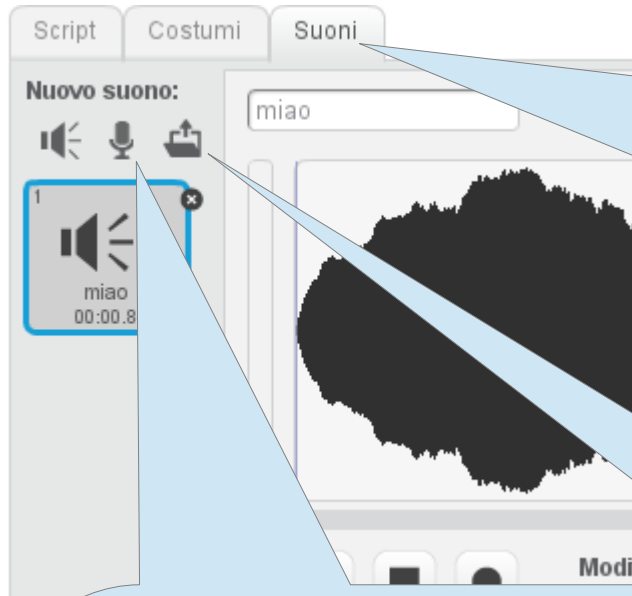
- **Suona tamburo** si possono scegliere diversi strumenti
- **Suona nota** c'è una tastiera con due ottave e nomenclatura anglosassone (A=La, B=Si, C=Do, ...) compresi # e b




ogni sprite ha un insieme di suoni da scegliere o estendere



# Registrazione suoni



- Nella linguetta **Suoni** nell'area di programmazione di uno sprite (o dello stage) c'è l'elenco dei suoni disponibili.
- Per esempio il gattino ha disponibile un **miao**, che posso ascoltare cliccando su 

- Se clicco su **[registra]** si apre il programma per registrare suoni del mio PC, e posso registrare.
  - Serve ovviamente un microfono (integrato o esterno)
  - **Attenzione:** su linux (ubuntu) a volte questo fa crashare scratch. Meglio salvare il progetto prima!

- Si possono anche importare suoni **[importa]**
  - Scratch ha una libreria di suoni già pronta, divisa per tipi
  - Oppure posso usare altri file audio (wav, mp3)

# Riassumendo: abbiamo imparato a



- **Creare programmi per uno sprite**
  - **Movimento**
  - **Costumi**
  - **Suoni**
  - **Interazione**
  - **cicli**
- **Gestire piu' di uno sprite**
- **Gestire sfondi**

# Proviamo a mettere tutto insieme



- **Un nuovo progetto:**
  1. **Proviamo a creare un gattino che cammina avanti e indietro per lo schermo, girandosi e miagolando quando arriva al bordo.**
  2. **Usiamo uno sfondo appropriato: anzi, piu' d'uno. E vogliamo poter scegliere tra 2 diversi sfondi usando le frecce destra e sinistra della tastiera**
    1. **Magari mettiamo anche una musica di sottofondo adeguata e diversa per i due sfondi.**
  3. **Infine, mettiamo un ballerino che danza sullo sfondo, che cambia colore ogni volta che il gattino tocca il bordo dello schermo**

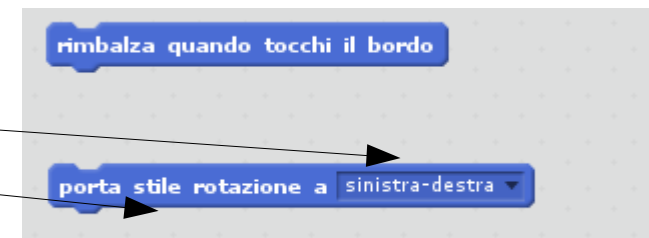
# Analisi del programma



## 1. Dovreste già saperlo fare.

1. Vi manca il fatto di girarsi quando si arriva al bordo, ma c'è un mattoncino apposta rimbalza quando tocchi il bordo
2. Attenzione che “rimbalzare” vuol dire non solo destra-sinistra, ma anche sopra-sotto, quindi al primo rimbalzo il gattino si troverà capovolto. Per evitarlo si usa il blocco “porta stile rotazione a [ ]”

1. Può ruotare
2. sinistra-destra
3. Non ruotare



## 2. Facile:

1. Unica cosa è comandare il cambio sfondo con i tasti freccia destra e sinistra
2. Possiamo far suonare la musica agli sfondi.
3. E' un po' più complicato: mancano alcuni pezzi

1. segue

# Azioni condizionali e coordinazione





- **Dovete far compiere ad uno sprite una azione quando qualcosa succede ad un altro sprite:**
- **scomponiamo il problema in due parti:**
  - **Uno sprite deve “parlare” all'altro sprite**
    - (“Hey, io ho toccato il bordo!),
  - **l'altro deve ascoltare e**
  - **quando sente il messaggio giusto fare quello che deve fare.**
    - **Messaggistica**
- **Il primo sprite deve parlare solo quando qualcosa accade:**
  - **azione condizionale**



# Inviare messaggi



- **Se ci sono due persone che devono coordinare una qualche attivita', bisogna che si parlino.**
  - Per esempio, quando il primo ha finito la sua parte, dice al secondo “ho finito” e allora il secondo puo iniziare a fare la sua parte.
- **Anche in programmazione e' cosi'.**
  - In scratch ogni sprite puo' inviare a tutti un messaggio usando  inserendo un nome per il messaggio (qui “bordo” )
  - Gli altri sprite ascoltano tramite 
    - Ovviamente devono “ascoltare” la stessa parola” che viene inviata dall'altro sprite (bordo)
    - I mattoncini che seguono questo controllo, vengono eseguiti dopo che e' arrivato il messaggio “bordo”

# Azione condizionale



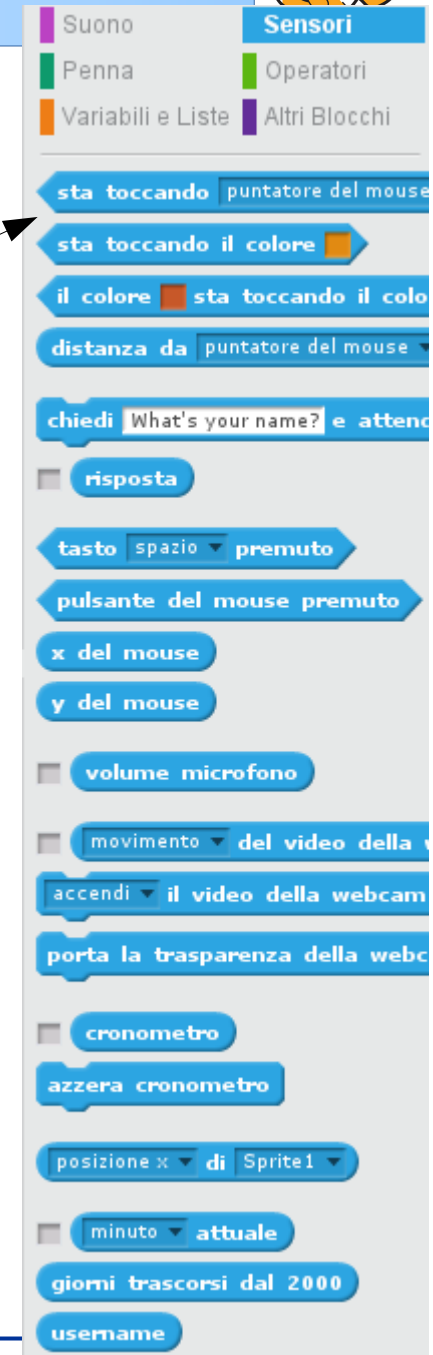
- **E' uno degli elementi base di qualunque programma. Devo fare qualcosa quando una certa condizione e' vera.**
- **Ci sono molti modi diversi per farlo.**
  - **Usiamo**
    - **Per sempre quando <succede qualcosa>**
  - **Da notare sempre la forma del mattoncino:**
    - **Puo' seguire o precedere qualsiasi altro mattoncino,**
    - **Tra le ganasce puo' accogliere uno o piu' mattoncini (come il ciclo per sempre)**
    - **C'e' un "buco" dopo il se con una forma particolare. Qui dobbiamo mettere la **condizione** che si dovra' verificare perche' le azioni tra le ganasce vengano eseguite.**



# Condizioni



- **Si trovano nella libreria [sensori]**
- **Hanno una forma particolare, con gli estremi spigolosi**
  - **<sta toccando [ ]>**
  - La forma e' quella adatta per inserirsi nel buco di **[se < >]**
  - Le opzioni di <sta toccando> sono [bordo, mouse, altri sprite]. Quello che ci serve.





- **Invece di rimbalzare e tornare indietro, immaginiamo che il nostro gattino cammini sempre in avanti,**
  - **Ogni volta che arriva al bordo destro:**
    - **Cambio sfondo a quello successivo**
    - **Riporto il gattino al bordo sinistro**
  - **Dopo aver camminato un po', si stufa e si ferma**
    - **Per esempio dopo 10 volte.**
      - **Complicato, serve un modo per contare quante volte ha raggiunto il bordo: vediamo dopo.**