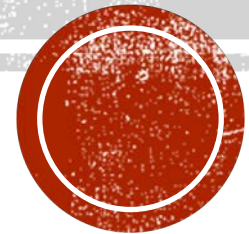


SCRATCH INCONTRA ARDUINO – I LEZIONE

M. Andreotti e G. Cibinetto (INFN Ferrara)

Lab2Go – 14 Gennaio 2021



Chi siamo

next.infn.it



Mirco Andreotti – Tecnologo INFN Ferrara
Fisico delle particelle (LHCb, BaBar, E835)
Docente di elettronica @ UNIFE
Esperto di sistemi di controllo e acquisizione



Gianluigi Cibinetto – Ricercatore INFN Ferrara
Fisico delle particelle (BESIII, BaBar, E835)
Docente di tecnologie di rivelazione @ UNIFE
Esperto di fisica adronica e rivelatori di particelle

Come nascono queste lezioni





Il rivelatore dell'esperimento BaBar

Indice delle lezioni

- Introduzione
 - Scratch: Elementi di Programmazione
 - Scratch: Esercitazioni di Laboratorio
- I lezione
- Arduino: Introduzione
 - Cenni di Elettronica per Arduino
 - Arduino: Esercitazioni di Laboratorio
 - Sommario
- II lezione
- Pensare in grande: approfondimenti e progetti
- III lezione?

Di cosa ci occuperemo

- **Scratch** e **Arduino**

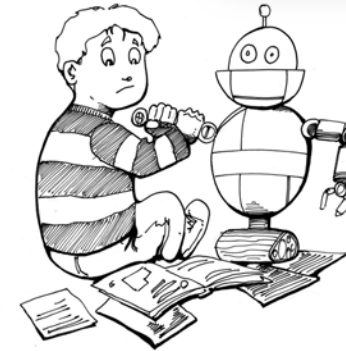
- Scratch è un Software
- Arduino è un pezzo di Hardware

- Cosa NON faremo in queste lezioni?

- NON vi insegneremo a programmare in Scratch
- NON vi insegneremo a utilizzare Arduino

- Allora cosa faremo?

- cosa si può **creare** con Scratch e Arduino, singolarmente e insieme
- **capire** perché è importante imparare queste materie indipendentemente dal percorso di studi o dagli interessi
- **idee** che potrete **sperimentare** da soli o con le vostre compagne e compagni



D'accordo, ma di che cosa stiamo parlando?

The Scratch logo, featuring the word "SCRATCH" in a stylized, orange, bubbly font with a blue outline.

- **Scratch** (in realtà noi useremo mBlock che è un suo derivato)
 - è un ambiente di programmazione, con un linguaggio di programmazione grafico.
 - Intuitivo, facile da usare, si possono avere buoni risultati in breve tempo, è gratuito
 - <https://scratch.mit.edu/>



- **Arduino**
 - è una piattaforma hardware composta da una serie di schede elettroniche dotate di un microcontrollore.
 - Intuitivo, facile da usare, si possono avere buoni risultati in breve tempo, è open source
 - <https://www.arduino.cc/>

- Perché questa scelta
 - **Intuitivi** e **facili** da usare
 - si possono avere buoni risultati in breve tempo con un budget limitato
 - Attenzione: facile da usare non vuol dire limitato



Obiettivi e competenze



- Perché insegnare Scratch e Arduino a scuola?

Cosa posso imparare?	A cosa mi serve?
Logica	La logica sta alla base della matematica, ma del ragionamento in generale
Parlare un'altra lingua	A conoscere nuovi mondi e nuove persone
Inserire figure nello spazio	Geometria, piano cartesiano
Utilizzo attivo della tecnologia	Produco, non subisco
Il valore dei numeri	Capire i dati e le statistiche
Pensiero critico	Confronto delle proprie idee maggiore apertura mentale
Risolvere problemi	A tutto
A divertirmi	A divertirmi... che non è poco!

Veniamo ai fatti: logistica del laboratorio (I)

Tempi

- L'impressione con questo genere di attività è che non ci sia mai tempo a sufficienza
- I tempi a scuole sono sempre molto ristretti e vincolati dall'orario generale delle lezioni: massimo due ore (talvolta da 50 minuti)
- Con un'ora a disposizione la cosa migliore sarebbe avere il laboratorio "personale" in modo da non dover smontare e rimontare tutto ogni volta
- Il tempo perso in una attenta preparazione e smontaggio dei componenti non è perso ma guadagnato
- Le attività di laboratorio che proponiamo hanno una durata di circa 90 minuti così strutturati (in appendice proponiamo anche come fare con 50 minuti a disposizione)



Durata*	Attività
10 min	introduzione ed esempi
10 min	progettazione dell'attività da parte degli studenti
50 min	esercizi di laboratorio
15 min	discussione dei risultati e sistemazione del laboratorio
5 min	riepilogo della lezione

* la durata delle singole attività è indicativa



Veniamo ai fatti: logistica del laboratorio (II)

Luoghi

- Un laboratorio di questo tipo spesso è vincolato alla disponibilità dei computer (i computer devono essere in buone condizioni, non occorre che abbiano grandi prestazioni, ma devono essere tenuti e usati responsabilmente)
- Meglio portatili che PC fissi (opinione personale)
- Non importa la distribuzione dei banchi purché ci sia spazio a sufficienza (ci deve essere una zona per **progettare**, una zona per **programmare**, una zona per **costruire**)
- I laboratori si fanno a piccoli gruppi... e si chiacchiera!



spesso sono piccoli dettagli che fanno la differenza, non l'utilizzo di strumentazione costosa. In altre parole siete voi, non i dirigenti scolastici







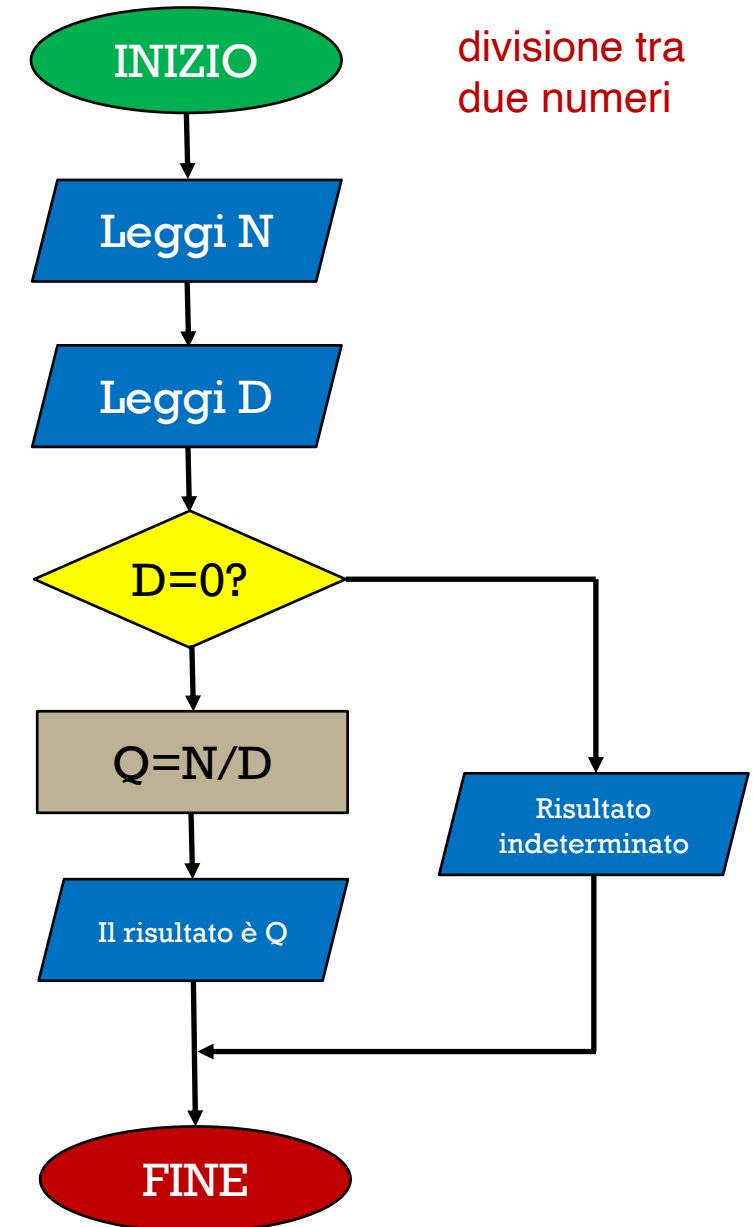
Scratch: Elementi di Programmazione

Programmi e algoritmi...

Un **programma** è un **software** che può essere eseguito da un **calcolatore**, che può ricevere certi dati (**input**) e che restituisce (**output**) una soluzione **algoritmica**.

Algoritmo → schema, procedimento, serie di istruzioni

SIMBOLO	SIGNIFICATO
	Inizio e fine della sequenza di istruzioni
	Inserimento ed emissione dei dati
	Istruzione da eseguire
	Istruzione che implica una scelta tra due possibili percorsi a seconda della valutazione di una certa condizione.



L'ambiente di sviluppo **SCRATCH**

The image shows the Scratch web interface. On the left is the 'Code' area with various block categories: Motion, Looks, Sound, Events, Control, Sensing, Operators, Variables, and My Blocks. A script is being built with the following blocks: 'when green flag clicked', a 'forever' loop containing 'move 10 steps' and 'if on edge, bounce'. The center stage shows a cat sprite. The right side features a 'Sprite' control panel with x and y coordinates set to 0, and a 'Stage' panel with a backdrop. Red arrows point from text labels to these elements: 'blocchi di istruzioni' points to the code blocks; 'questo è un pezzo di programma' points to the script; 'questo è il risultato del programma' points to the cat sprite; 'questo è uno sprite' points to the cat icon in the sprite panel; 'esegue/interrompe il programma' points to the green flag icon; 'sistema di coordinate cartesiane' points to the x and y coordinate fields; and 'qui si modificano gli "sprite"' points to the sprite panel.

blocchi di istruzioni

questo è un pezzo di programma

esegue/interrompe il programma

questo è il risultato del programma

questo è uno sprite

sistema di coordinate cartesiane

qui si modificano gli "sprite"

In realtà noi useremo **mBlock** (<https://ide.mblock.cc>)

questo è il risultato del programma

The screenshot shows the mBlock IDE interface. At the top, there's a blue header with 'mBlock', 'File', 'Edit', 'Untitled', and 'Save'. On the right, there are 'Tutorials', 'Feedback', and a user profile icon. The main workspace is divided into three sections: a stage with a panda sprite, a block palette, and a script area. The block palette has categories: Motion, Looks, Sound, Events, Control, Sensing, Operators, and extension. The script area contains a 'when clicked' event block, a 'forever' loop containing 'move 10 steps' and 'if on edge, bounce' blocks. The 'if on edge, bounce' block is highlighted with a red arrow. The 'Sprites' tab is active, showing a 'Panda' sprite with its properties (X: 0, Y: 0, Size: 100, Direction: 90) and an 'add' button. A red arrow points to the 'add' button with the text 'qui si modificano gli "sprite"'. Another red arrow points to the 'when clicked' block with the text 'questo è un pezzo di programma'. A third red arrow points to the 'if on edge, bounce' block with the text 'blocchi di istruzioni'. A fourth red arrow points to the 'Panda' sprite in the 'Sprites' tab with the text 'Tab per l'inserimento di device (e.g. Arduino)'.

mBlock ha una serie di funzionalità già presenti per la programmazione di Arduino

La struttura di un programma

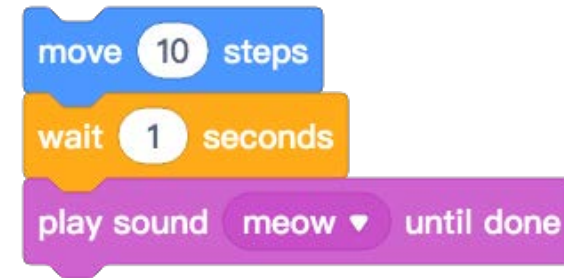
▪ Inizio

- L'inizio del programma è generalmente (sempre nel nostro caso) comandato da noi



▪ Esecuzione

- È quello che fa il programma, la parte divertente



▪ Fine

- Programmi che finiscono da soli (e.g. la riproduzione di un video)
- Programmi che finiscono perché vengono interrotti (e.g. finisco di scrivere un documento e chiudo)
- Programmi che finiscono in seguito a qualcosa che accade (e.g. finisco le vite di Super Mario Bros.)

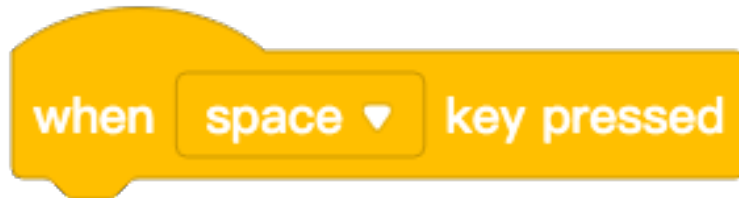


Eventi

sono istruzioni che servono a fare partire una sequenza



quando viene cliccata la bandierina verde



quando viene premuta la barra spaziatrice

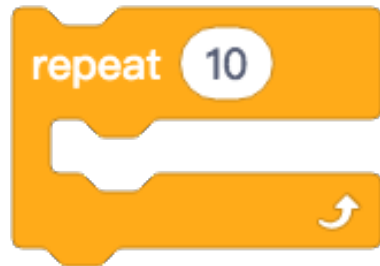


quando questo sprite viene cliccato

Controlli

consentono l'esecuzione di una sequenza di istruzioni al verificarsi una condizione

Cicli



ripete
10 volte

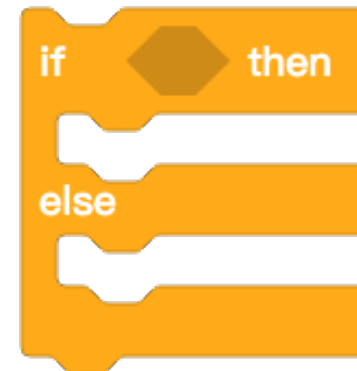


ripete
all'infinito

Condizioni



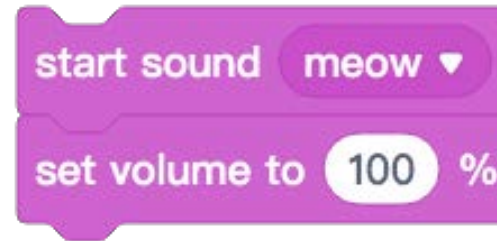
se condizione
allora istruzioni



se condizione
allora istruzioni
altrimenti altre istruzioni

Output

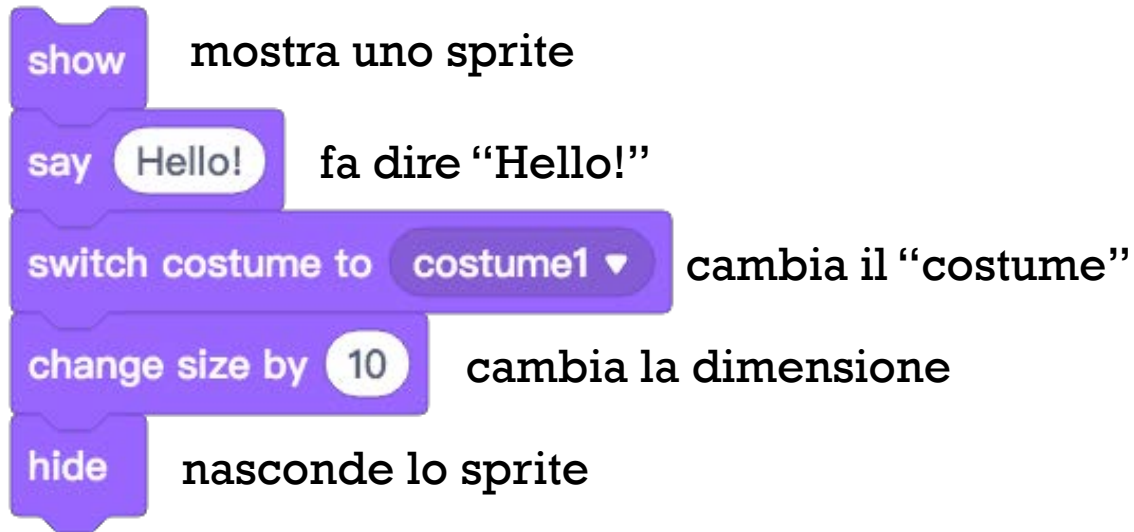
Sound



riproduce un suono

imposta il volume

Looks



mostra uno sprite

fa dire "Hello!"

cambia il "costume"

cambia la dimensione

nasconde lo sprite

Motion



posiziona lo sprite al punto (117;0)

muoviti di 10 passi

posizionati in direzione 90 gradi

cambia la x di 10

ruota in senso orario di 15 gradi

Variabili e operatori

Make a Variable ← crea una variabile

- a
- b
- media_voti
- ViteDiMario
- X

set a ▼ to 0 imposta il valore della variabile "a"

change a ▼ by 1

media_voti > 8 maggiore

X + 100 somma

pick random 1 to 10 restituisce un numero casuale

apple contains a ? restituisce "true" se la parola "apple" contiene una "a"

join apple banana unisce le parole "apple" e "banana"

letter 1 of apple restituisce la prima lettera della parola "apple"

esempio →

when clicked

set a ▼ to Ciao

say a

Input

- touching **Panda** ? lo sprite tocca il Panda
- touching **color** ? lo sprite tocca il colore azzurro

- key **space** pressed? viene premuto il tasto spazio
- key **right arrow** pressed? viene premuta la freccia destra
- key **any** pressed? viene premuto un qualsiasi tasto

```
ask What's your name? and wait
```

fa una domanda, ne aspetta la risposta il cui valore viene attribuito alla variabile

answer

```
ask What's your name? and wait
say join ciao answer
```

```
if touching color red ? then
  say Rosso
```

```
repeat until key any pressed?
  start sound pop
```

spesso utilizzati come condizioni nei controlli

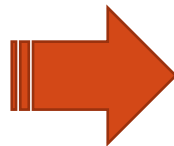


Scratch: Esercitazioni di Laboratorio

Esercizio 1: Iniziamo a muovere gli Sprite

- Muoversi a destra e sinistra e saltare.

```
when green flag clicked
  go to x: 0 y: 0
  forever loop
    if key right arrow pressed? then
      change x by 2
    if key left arrow pressed? then
      change x by -2
```



```
when green flag clicked
  go to x: 0 y: 0
  set rotation style left-right
  forever loop
    if key right arrow pressed? then
      point in direction 90
      change x by 2
    if key left arrow pressed? then
      point in direction -90
      change x by -2
```



```
when green flag clicked
  forever loop
    if key space pressed? then
      repeat 10
        change y by 3
      repeat 10
        change y by -3
```

23

Alt! come si
controllano gli errori
(senza impazzire)?

Esercizio 2: variazioni sul tema

- La camminata
 - Cerchiamo di far camminare e saltare il nostro personaggio
 - File: 02_walk.mblock

Esercizio 3: interazione tra Sprites

- Touching
 - Ora facciamo interagire due Sprites tramite il tatto.
 - File: 03_ilMioPrimoGioco.mblock

Esercizio 4: interazione tra Sprites

- Broadcast message
 - Ora invece creiamo un'interazione a distanza attraverso l'emissione di un messaggio
 - File: walkBee1.mblock
 - File: walkBee2.mblock

Esercizio 5: clonare uno Sprite

- Il barone rosso
 - Può essere utile fare molte copie di uno Sprite senza dover per forza ripetere le stesse istruzioni
 - File: 05_airplane.mblock

Esercizio 6 e 7: giochi di matematica

- Calcolare la media di una serie di numeri
 - Calcolare la media di una serie di numeri inseriti manualmente
 - File: 06_media.mblock
- Piano cartesiano
 - Giocare con le coordinate cartesiane
 - File 07_pianoCartesiano.mblock

Esercizio 8 e 9: statistica e analisi dati

- Fare grafici
 - Grafici di grandezze misurate o casuali
 - File: 08_statistica.mblock
- Collegamento con Sheet di google drive
 - Salvare i dati acquisiti o generati su un file Excel
 - File: 09_excel.mblock

Musica, Arte e Lingue Straniere

- Con mBlock è possibile utilizzare librerie aggiuntive per rendere le attività ancora più interessanti e coinvolgenti.
 - Con le librerie di musica è semplicissimo creare ritmi e giochi sonori
 - La libreria Pen vi darà la possibilità di disegnare nei vostri programmi
 - Ci sono Librerie che, connesse a internet, rendono possibile la traduzione dei vocaboli da una lingua all'altra
 - E' possibile fisicamente interagire con il programma attraverso la libreria di Video Sensing
- Infine, ed è la cosa per cui siamo qui, possiamo programmare microcontrollori tipo Arduino



Riassumendo

Sommario

- Scratch è una delle più belle invenzioni per imparare a programmare
 - E' facile, intuitivo, può essere insegnato con profitto anche a chi non sa ancora leggere e scrivere bene
 - Aiuta a formare nella mente gli algoritmi di cui la matematica è piena così come la vita
- L'apprendimento va fatto con luoghi, tempi e modalità proprie, che non sono quelli della didattica standard
 - Abbiamo parlato di come si passa da un algoritmo a un programma, di come si controllano gli errori
 - Abbiamo mostrato diverse possibilità per iniziare e qualcosa di più complesso
 - Abbiamo accennato a possibilità di interazione tra musica, disegno, lingue, eccetera
- **La cosa più importante a questo punto siete voi, questo è un potentissimo strumento per voi: il limite ora è la vostra fantasia!**