

**TEMA**

# Coding a blocchi o testuale?

**Antonio Faccioli**

[antonio.faccioli@soluzioniopen.com](mailto:antonio.faccioli@soluzioniopen.com)

**keywords:** : *coding, scratch, logo, librelogo, libreoffice, geometria della tartaruga*

## Introduzione

In questo articolo non mi fermerò a trattare cosa sia il pensiero computazionale oppure cosa si intenda per *coding*, pertanto darò queste terminologie come già in possesso del lettore.

Oggetto dell'analisi sarà invece l'uso didattico di linguaggi di programmazione visuali a blocchi e testuali. In particolare metterò a confronto il più conosciuto [Scratch](#) con [LibreLogo](#), la variante di [LOGO](#) contenuta all'interno di LibreOffice.

## Linguaggi a blocchi

In questa tipologia di linguaggi la programmazione avviene all'interno di un'applicazione, che permette di interpretare ed eseguire il "codice" realizzato tramite la disposizione sequenziale di blocchi.

Il più conosciuto di questa tipologia è senz'altro Scratch, ma vi sono molte altre soluzioni. Su Wikipedia è disponibile un [esaustivo elenco](#) con tutti i riferimenti.

Questi linguaggi consentono di creare giochi, animazioni, storie interattive, sequenze musicali oppure possono essere usati per impartire comandi ad un robot. In questo modo si può programmare senza scrivere una sola riga di codice, in quanto il tutto avviene attraverso l'ordinamento e lo spostamento di una serie di oggetti grafici, disponendoli in sequenze. Ad ogni blocco corrisponde un'istruzione che permette di eseguire un comando che farà muovere un robot o un personaggio a video, accendere dei LED su una scheda o far cambiare il colore ad un oggetto.

I punti di forza di Scratch sono sicuramente la sua gratuità, l'interfaccia studiata per un utilizzo facilitato e la possibilità, una volta che ci si è registrati, di condividere i propri lavori attraverso la piattaforma, ma anche di poter studiare e rielaborare il lavoro di altri.

## Linguaggi testuali

Questi sono sicuramente, nell'immaginario collettivo, i ferri del mestiere degli informatici professionisti. In questo caso la programmazione avviene attraverso la scrittura di codice utilizzando degli editor di testo, che possono però semplificare la digitazione utilizzando ad esempio l'autocompletamento dei comandi, come avviene nei programmi di videoscrittura.

Le istruzioni non sono oggetti grafici o blocchi, ma vocaboli (generalmente inglesi) scritti in sequenze e suddivisi in uno o più file che verranno compilati o interpretati per essere eseguiti. La compilazione è un processo che permette di creare un file eseguibile (ad esempio *exe* o *msi* in ambiente Windows). L'interpretazione invece è una soluzione "simile" a quella dei visuali, ovvero il codice viene direttamente eseguito da un'applicazione (Interprete).

Tra i linguaggi testuali troviamo che alcuni sono nati appositamente per l'uso didattico oppure alcuni professionali che per la loro duttilità possono essere utilizzati anche per laboratori di coding a scuola. Nel primo caso possiamo citare LOGO, nel secondo [Python](#).

Python, fu sviluppato agli inizi degli anni novanta e può essere utilizzato attraverso degli interpreti. I suoi punti di forza sono la dinamicità, semplicità e flessibilità nonché la velocità. E' un linguaggio multiplatforma e quindi utilizzabile su Windows, OSX e Linux.

Grazie alla sua semplicità, nonché ad alcune librerie sviluppate appositamente, Python si presta bene ad essere utilizzato in un ambiente didattico e formativo.

LOGO venne creato da Wally Feurzeig, Cynthia Solomon e Seymour Papert. Quest'ultimo come molti sapranno fu il primo teorizzatore del pensiero computazionale ed ideò questo linguaggio proprio per utilizzare il computer con i bambini della scuola primaria.

## Testuali VS Blocchi

Il vantaggio dei sistemi visuali è senza ombra di dubbio la possibilità di programmare senza preoccuparsi della sintassi, avendo quindi la possibilità di concentrarsi sull'obiettivo, di apprendere quelli che possono essere alcuni dei costrutti base. Imparare ad esempio come funzionano alcuni cicli, le condizioni, le variabili o le liste.

Tuttavia a lungo andare ed approfondendo l'uso della programmazione questi linguaggi mostrano i propri limiti: la rigidità e la conseguente poca flessibilità.

I testuali invece sono più complessi, necessitano dello studio della sintassi. Le istruzioni devono essere scritte correttamente, gli errori di digitazione possono portare a problemi nell'esecuzione. Questi linguaggi peraltro offrono una maggiore flessibilità e potenzialità di espansione che i blocchi ben definiti non possono dare. Inoltre il codice scritto può essere letto, come si farebbe con un linguaggio naturale, molto più semplicemente rispetto ad un sistema visuale.

## Alcuni esempi di utilizzo

Considerando che lo sviluppo dei linguaggi di programmazione didattici è passato dal testuale al visuale e considerato che i sistemi di sviluppo professionale invece si basano ancora per la stragrande maggioranza sull'utilizzo dei testuali, ho sperimentato in alcuni corsi e laboratori a scuola l'accostamento delle due tipologie.

Il lavoro fatto, sia in corsi con docenti sia con alunni della scuola primaria, è stato di partire con attività di *coding* utilizzando Scratch e successivamente passare a soluzioni testuali con LibreLogo. Quest'ultimo è una implementazione del linguaggio LOGO all'interno di LibreOffice. La scelta è ricaduta su questo "dialetto" perché per utilizzarlo si passa attraverso ad uno strumento familiare come Writer, un programma di videoscrittura.

Il percorso è stato incentrato sulla costruzione di figure geometriche attraverso i due linguaggi, prendendo spunto dalla [geometria della tartaruga](#) ma andando a studiare diversi costrutti come cicli, variabili e condizioni.

Utilizzando in Scratch i blocchi situazioni, controlli, movimento e penna si sono studiate le sequenze per costruire quadrati, triangoli nonché la ripetizione di queste figure piane per creare nuove forme più o meno complesse.

Una volta appresa la sequenza di istruzioni a blocchi, si passava alla scrittura del codice in LibreLogo. Ripetendo pari pari quanto fatto in Scratch e successivamente ampliando questi script con istruzioni proprie del linguaggio testuale, come il riempimento delle aree.

In un laboratorio con ragazzi di una classe quinta primaria abbiamo ad esempio creato in Scratch uno script per disegnare un fiocco di neve e successivamente abbiamo realizzato il lavoro con LibreLogo.

- Un esempio dell'attività svolta è presente in questi [video](#) disponibili su Youtube:

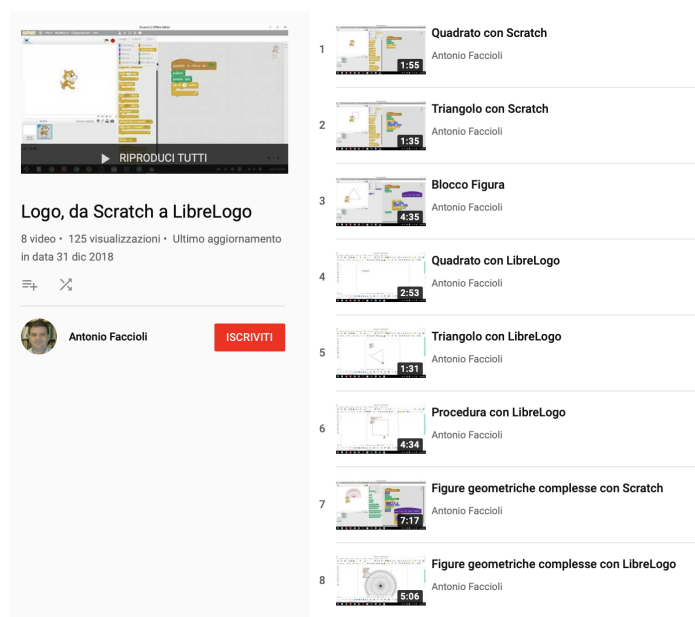


Figura 1 - Video sul disegno di figure geometriche con Scratch e con LibreLogo

Un ulteriore esempio di lavoro di raffronto (più articolato) è l'albero di Pitagora realizzato con Scratch ([script](#) e [video](#)) e LibreLogo ([script](#) e [video](#)).

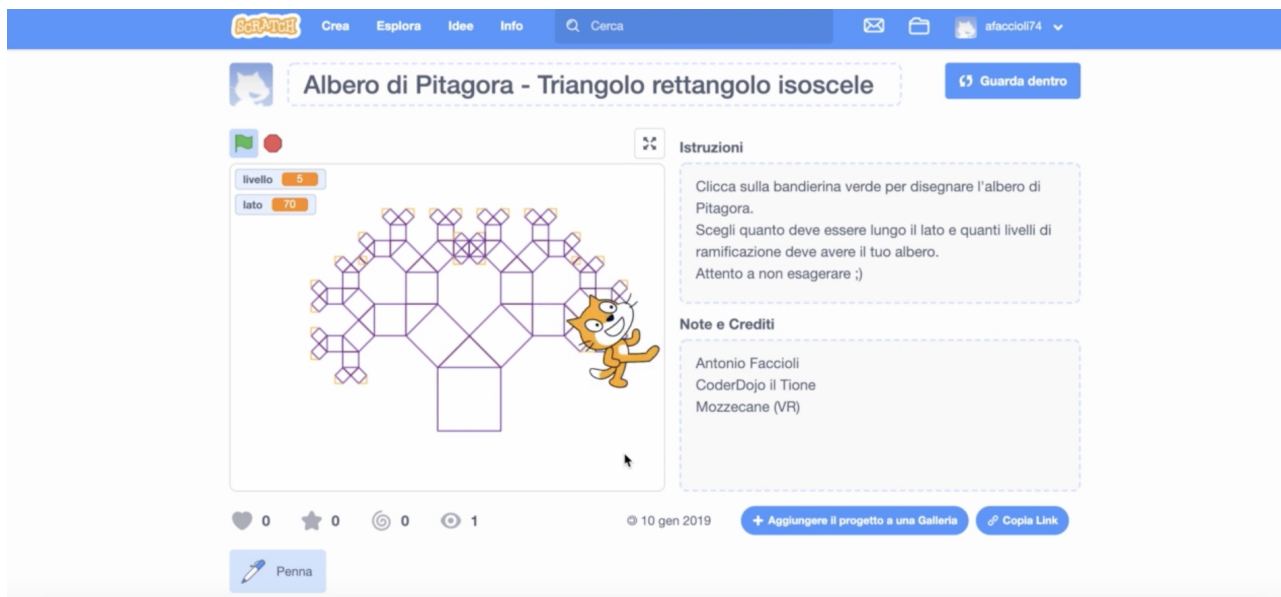


Figura 2 - L'albero di Pitagora disegnato con Scratch.

## Conclusioni

In tutte le situazioni sperimentate la percezione dei partecipanti è stata dapprima di una leggera difficoltà dovuta alla scrittura e alla sintassi, tuttavia superabile facilmente perché le sequenze di istruzioni erano già acquisite e quindi diveniva più semplice concentrarsi sulla scrittura.

Una volta appresa la sintassi, i partecipanti (in particolare i docenti) apprezzavano il minor numero di istruzioni testuali rispetto ai blocchi e la possibilità poi di maggiori opportunità date dalle istruzioni a disposizione.

E' da sottolineare che il confronto è stato fatto solo con questa tipologia di elaborati e abbiamo evidenziato che per molti aspetti Scratch rimaneva l'ideale in situazioni in cui si voleva realizzare storytelling o semplici videogiochi. Naturalmente in quest'ultimo caso, per un opportuno raffronto, avremmo dovuto utilizzare altri linguaggi testuali (ad es. Python), mettendo in gioco altre difficoltà.

Tuttavia i docenti evidenziavano stimolante ed interessante la strada di partire da sistemi visuali per poi passare a linguaggi testuali, proseguendo nella maturazione di competenze in questa tipologia (ad esempio Scratch, LibreLogo e Python).