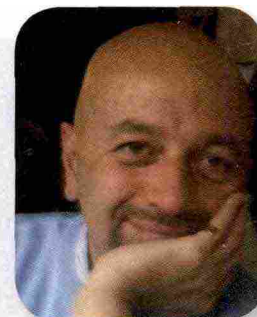


Editoriale

Coding Time

di Pier Cesare Rivoltella, Università Cattolica del Sacro Cuore di Milano



Negli ultimi mesi il CREMIT, il centro di ricerca che dirigo in Università Cattolica, si è trovato coinvolto in alcuni progetti di ricerca e formazione che ruotano attorno al Coding. Penso in particolare alla ricerca *SMART Coding*, condotta insieme all'INDIRE e diretta a verificare le rappresentazioni degli insegnanti, degli studenti e dei loro genitori in materia di coding, e al progetto *It's Coding time* che vede coinvolta una grossa rete di scuole della provincia di Parma e nel quale il CREMIT è impegnato, insieme al Servizio Marconi dell'USR Emilia Romagna e alla cooperativa *Officine on/off* di Parma, in un doppio percorso, di formazione degli insegnanti e ancora una volta di ricerca sul senso e l'efficacia del fare coding in classe.

Come si ricorderà, il coding, insieme allo sviluppo della creatività, al ruolo dell'arte e della musica, era uno dei temi in primo piano nel documento sulla "Buona Scuola" che legava a esso la possibilità per la scuola di fare innovazione e di avvicinare i suoi linguaggi e le sue pratiche a quelli del mondo degli studenti. Ma che cos'è il coding?

Da Scratch ai FabLab

Coding, se lo traducessimo letteralmente dall'inglese, vorrebbe dire "fare codice". Ovvero: programmare, scrivere codice, usare il linguaggio della macchina per fornirle istruzioni e farla operare secondo le nostre intenzioni. La sua diffusione, in Italia, è legata alla nascita dei primi *Coderdojo* nel 2012, degli spazi creativi la cui finalità era ed è la diffusione gratuita della cultura della programmazione, soprattutto ai più piccoli. Tra le attività da essi promosse vi sono corsi di programmazione in *HTML5* e *Javascript* (due dei linguaggi più diffusi), corsi sull'uso di *Arduino*, una scheda elettronica programmabile che consente di sviluppare microrealizzazioni (e infatti, anche con i bambini, una delle applicazioni più interessanti del coding è proprio la realizzazione di piccoli robot), l'uso di *Scratch*.

Scratch (cfr. in Internet, URL: <https://scratch.mit.edu/>) è un linguaggio di programmazione a oggetti ispirato alla teoria costruzionista e sviluppato da Mitchel Resnick presso il MediaLab del MIT di Boston. La finalità del programma è di consentire anche ai bambini di programmare operando su un linguaggio molto visuale che ha alla base la metafora del gioco di costruzioni. Come nel LEGO, si tratta di mettere insieme dei blocchi nel giusto ordine.

Tutte queste attività sono spesso parte integrante dei *FabLab*, dei luoghi creativi che si stanno diffondendo rapidamente nelle nostre realtà urbane e che si pensano come uno spazio in cui sia possibile l'aggregazione giovanile, il co-working e lo sviluppo d'impresa. In un *FabLab* si fa naturalmente coding, ma oltre a questo si dispone di stampanti 3D grazie alle quali realizzare materialmente i propri progetti e di altre macchine, come le *lascutter*, grazie alle quali realizzare gli stessi oggetti facendoli ritagliare e scolpire dal laser a partire da un materiale dato. Molti *FabLab* entrano in contatto con le scuole, le accompagnano nei percorsi di coding e poi consentono ai bambini di produrre materialmente le loro creazioni.

Prepararsi al futuro e pensare con stile

A cosa può servire portare in scuola tutto questo? E alla primaria, in particolare? Si può rispondere rifacendosi ai risultati di una ricerca che alcuni studiosi finlandesi hanno condotto sulle motivazioni che stanno alla base del coding e della sua promozione nelle scuole.

La prima di queste motivazioni è *funzionalistica*. In una società dell'informazione come la nostra, in cui la rivoluzione digitale e la diffusione delle tecnologie dell'informazione hanno pervaso e improntano di sé praticamente qualsiasi ambiente di vita e professione, fare coding già alla scuola primaria significa iniziare a conoscere e usare i linguaggi che in futuro consentiranno al bambino un migliore inserimento nel mondo del lavoro.

Una seconda motivazione è *espressiva*. Imparare un linguaggio significa sempre poterlo poi usare in maniera creativa. Il coding andrebbe dunque legato allo sviluppo della creatività del bambino, servirebbe a fornirgli un ulteriore elemento per liberare le sue possibilità comunicative. Dentro una "scuola del fare" il coding è la versione aggiornata di tutti quei materiali e di tutti quei linguaggi grazie ai quali si è sempre favorito l'avvicinamento del bambino alla cultura materiale: il coding, oggi, come la tipografia al tempo di Freinet.

I sostenitori del coding dicono a questo riguardo che esso favorirebbe lo sviluppo del pensiero computazionale, alludendo al fatto che il nostro cervello lavorerebbe in questi termini, cioè costruendo e applicando algoritmi alle sue scelte di azione. Ora, la recente ricerca neuroscientifica ha dimostrato che il modello di comprensione di come lavora il nostro cervello è l'*embodiment* più che il pensiero computazionale: noi funzioniamo in maniera adattiva e complessa, siamo un organismo e non un computer. Ma certo la capacità di costruire e utilizzare algoritmi (cioè mettere in fila le sequenze di operazioni che ci possono consentire di svolgere un compito complesso) è importante. Anche senza le macchine è possibile lavorarci già dalla scuola dell'infanzia: insegnando al bambino a comprendere ed eseguire ordini, abituardolo a risolvere problemi pianificando percorsi, favorendone l'acquisizione di routines. Fare coding insegna a pensare, a pensare con stile.

Dietro le interfacce

Vi sono ancora due motivazioni alla base del coding. Una è legata alla sua funzione *interpretativa*. Lo sviluppo dell'informatica di consumo è legato alle interfacce grafiche. Quando siamo davanti al nostro computer e con un clic sull'icona del formato cambiamo la giustificazione del testo sullo schermo, con un gesto estremamente semplice abbiamo ottenuto un risultato che dal punto di vista delle istruzioni date alla macchina è molto complesso. Gli americani esprimono tutto questo dicendo che "what you see is what you get", "quel che vedi è quello che ottieni". Noi non sappiamo cosa sia successo "dietro" all'interfaccia: ci basta sapere che quel clic produce quel risultato. Bene, imparare a conoscere i linguaggi significa capire a cosa corrisponda quel clic: significa sviluppare consapevolezza e, con essa, senso critico.

E arriviamo così all'ultimo significato del coding, un significato *emancipatorio*. Analisi e sintesi, smontare e rimontare, sono da sempre le operazioni-chiave del pensiero occidentale. Impararle significa emanciparsi dal rischio di essere controllati. Questo vale anche nel mondo dei media digitali, un mondo nel quale come dicevamo spesso rinunciamo al controllo in cambio della facilità. Certo, se all'impaginazione del mio testo ci pensa il programma da solo, per me è più comodo, ma in questo modo mi sto adeguando alle scelte che il programma ha fatto al mio posto. Conoscere il codice vuol dire, invece, poter fare diversamente.

A queste istanze – smontare i messaggi, pensare autonomamente – sono legati i temi della cittadinanza attiva e del media-attivismo. Il valore ultimo del coding sta proprio in questo suo potenziale educativo: fare coding significa sviluppare la *lifeskill* del pensiero critico, significa fare Media Education.