

*Estratto da*

## **Lifelong Kindergarten: Cultivating Creativity through Projects, Passion, Peers, and Play**

Mitchel Resnick, MIT Media Lab  
Pubblicato da MIT Press (2017)

Tradotto da Raffaella Micheli, Augusto Chiocciariello e il favoloso team di LCL Italia  
Revisione di Carmelo Presicce

### **Creatori di cose**

Nel gennaio del 2009 stavo guardando l'insediamento di Barack Obama come 44esimo presidente degli Stati Uniti in un'aula magna del MIT. La sala era gremita con più di 500 persone, e il video del discorso di insediamento di Obama veniva proiettato su due grandi schermi. Dato che tra il pubblico c'erano molti scienziati e ingegneri del MIT, non c'è da sorprendersi se la reazione più forte avvenne quando Obama dichiarò: "Riporteremo la scienza al posto che le spetta". Nella sala scoppiò un applauso.

Ma non fu quella la frase del discorso inaugurale che catturò la mia attenzione. Per me, il momento più memorabile fu quando Obama disse: "Sono stati coloro che hanno saputo osare, che hanno agito, coloro che hanno creato cose – alcuni riconosciuti, ma più spesso uomini e donne rimasti nell'ombra con il loro lavoro - che hanno portato avanti il lungo, accidentato cammino verso la prosperità e la libertà."

Coloro che osano, che agiscono, che creano cose. Questi sono gli studenti X, i pensatori creativi. Sono stati il motore del cambiamento economico, tecnologico, politico e culturale attraverso la storia. Oggi, tutti hanno bisogno di essere audaci, di mettersi in gioco, di essere costruttori — non necessariamente per piegare il corso della storia, ma per essere artefici delle proprie vite.

Usando la frase "coloro che creano cose" (*N.d.T.*, "*makers of things*"), Obama stava facendo un riferimento implicito a un movimento che stava appena iniziando a diffondersi nella nostra cultura: il Movimento Maker. Quest'ultimo cominciò come movimento popolare, in scantinati, garage, e community center tra persone che avevano la passione di fare cose, e di condividere le loro idee e creazioni gli uni con gli altri.

Nel 2005 il movimento guadagnò slancio con l'uscita della rivista Make di Dale Dougherty, che celebrava la gioia del costruire, creare e inventare cose. Lo scopo della rivista era di rendere alla portata di tutti la capacità di costruire, mostrando come chiunque potesse appassionarsi in attività fai da te. Il primo numero descriveva "cose straordinarie che persone comuni fanno nei loro cortili e garage", fornendo istruzioni per costruire un aquilone per fare foto aeree, un involucro termoelettrico per mantenere la birra fresca, e lightstick per creare giochi di luce dinamici la notte.

L'anno successivo, nel 2006, Dale organizzò la prima Maker Faire, definendola un "festival delle invenzioni, della creatività e dell'intraprendenza, adatto alle famiglie". C'erano esposizioni e laboratori per gioielleria, falegnameria, robotica — e per costruire quasi tutto quello che si possa immaginare. Negli ultimi dieci anni, centinaia di Maker Faire sono spuntate in tutto il mondo, attraendo milioni di ingegneri, artisti, designer, imprenditori, educatori, genitori e bambini.

Per molte persone, l'attrattiva del Movimento Maker è nella tecnologia. C'è stata una proliferazione di nuove tecnologie, come le stampanti 3D e il taglio laser, che ha reso possibile alle persone di progettare, produrre e personalizzare oggetti fisici. Molti sono affascinati dal potenziale economico di queste tecnologie, prevedendo che il Movimento Maker scatenerà una nuova rivoluzione industriale, nella quale le piccole imprese (o perfino quelle individuali) potranno fabbricare prodotti che precedentemente richiedevano grosse aziende con economie di scala.

Io sono attratto dal Movimento Maker per ragioni diverse. Credo che abbia il potenziale per essere non solo un movimento tecnologico ed economico, ma anche un movimento educativo, che fornisce alle persone nuovi modi di partecipare in esperienze di apprendimento creativo. Quando le persone fanno e creano, hanno l'opportunità di crescere come pensatori creativi. Dopo tutto, fare è alla radice della creatività.

Forse la cosa più importante è che il Movimento Maker incoraggia le persone a lavorare su progetti — la prima delle quattro P dell'apprendimento creativo. Gli articoli nella rivista Make e le mostre alla Maker Faire non insegnano soltanto le tecniche per creare; supportano un approccio all'apprendimento basato su progetti, in cui le persone imparano nuove idee, competenze e strategie mentre lavorano a progetti personalmente significativi. Dale Dougherty si riferisce ai progetti come "l'unità di base del *making*".

Crescendo, ho sperimentato personalmente il potere dei progetti. Da bambino mi divertivo a praticare tutti i tipi di sport: baseball, basket, tennis e altri. Ma più che praticare sport, mi divertivo a “creare” sport. Inventavo continuamente nuovi sport da giocare con mio fratello e mio cugino. Avevo la fortuna di avere un cortile per costruire e giocare — e la fortuna di avere genitori che mi permettevano di trasformare il cortile in un laboratorio per i miei progetti.

Un'estate, scavai nel giardino per creare il mio campo da golf in miniatura. Era un'esperienza di apprendimento continuo. Cominciai scavando semplici buche nel terreno per fare quelle da golf, ma scoprii che col tempo le buche perdevano la loro forma, così iniziai a rivestirle con lattine di alluminio. Funzionò finché non piovve, e le lattine si riempirono di acqua che era difficile da togliere. La mia soluzione fu di rimuovere anche il fondo della lattina prima di inserirla nella buca, in modo che il fondo potesse drenare l'acqua.

Quando aggiunsi pareti e ostacoli nel mio mini campo da golf, dovetti capire in che modo la palla avrebbe rimbalzato quando li urtava. Questo mi fornì lo spunto per imparare la fisica delle collisioni. Trascorsi ore a calcolare e a misurare gli angoli in modo da poter far rimbalzare la palla sugli ostacoli e dentro la buca. Quell'esperienza fu più memorabile di qualsiasi altra lezione di scienze avessi mai avuto in classe.

Lungo il percorso, cominciai a sviluppare consapevolezza non solo sul procedimento per creare un campo da golf in miniatura, ma sul processo generale per creare qualsiasi cosa: come cominciare con un'idea iniziale, sviluppare un piano d'azione preliminare, creare una prima versione, provarla, chiedere ad altri di provarla, rivedere il piano in base a cosa è successo — e continuare a ripetere tutto questo più volte. Lavorando al mio progetto stavo acquisendo esperienza con la Spirale dell'Apprendimento Creativo.

Attraverso questo tipo di progetti, iniziai a vedere me stesso come qualcuno che poteva fare e creare cose. Iniziai a guardare alle cose che mi circondavano in un modo nuovo, fantasticando su come fossero state fatte. Com'è fatta una pallina da golf o una mazza? Iniziai a immaginare quali altre cose avrei potuto creare.

Facendo una ricerca sul sito di Make ([makezine.com](http://makezine.com)), si trovano molti articoli che descrivono progetti di campi da golf in miniatura, dai titoli come “Costruisci da solo il tuo Golf da Tavolo” e “Golf Urbano in Miniatura 2.0”. Le tecnologie si sono evolute da quando costruì il mio campo da golf in miniatura, circa 50 anni fa. Adesso è possibile produrre ostacoli personalizzati con una stampante 3D o il taglio laser, ed è possibile

integrare sensori negli ostacoli, attivando motori o LED che si accendono non appena la palla rimbalza su un ostacolo.

Sono ancora molto orgoglioso del mio campo da golf in miniatura vecchio stile che ho costruito da ragazzo. Ma sono anche entusiasta che le nuove tecnologie possano ampliare i tipi di progetti che i ragazzi possono creare — e ispirare altri a diventare creatori di cose.

## Imparare costruendo

Nel corso degli anni, molti educatori e ricercatori hanno promosso il *learning by doing* (N.d.T. “*imparare facendo*”) sostenendo che le persone imparano meglio quando sono coinvolte nel fare cose, imparando attraverso attività pratiche.

Ma nella cultura del Movimento Maker, però, non è sufficiente fare qualcosa: bisogna costruire qualcosa. Secondo l’etica maker, le esperienze di apprendimento più preziose sopraggiungono quando si è attivamente impegnati nel progettare, costruire, o creare qualcosa — quando si impara costruendo (N.d.T. “*learning by making*”).

Per capire meglio la connessione tra costruire e imparare, e sul come promuovere l’apprendimento attraverso il costruire, non c’è posto migliore a cui volgere lo sguardo che il lavoro di Seymour Papert. Ho avuto la fortuna di lavorare con Seymour per molti anni al MIT. Seymour ha sviluppato, più di qualsiasi altro, le basi intellettuali per imparare costruendo, insieme a tecnologie e strategie avvincenti per sostenerlo. In realtà, Seymour dovrebbe essere considerato il santo patrono del Movimento Maker.

Seymour amava l’apprendimento in tutti i suoi aspetti: comprenderlo, sostenerlo, farlo. Dopo aver ottenuto un dottorato in matematica dall’Università di Cambridge nel 1959, Seymour si trasferì a Ginevra per lavorare con il grande psicologo svizzero Jean Piaget. Attraverso un’attenta osservazione e interviste con migliaia di bambini, Piaget scoprì che i bambini costruiscono attivamente la conoscenza attraverso le loro interazioni quotidiane con persone e oggetti che li circondano. La conoscenza non è infusa nei bambini come l’acqua in un vaso. Al contrario, i bambini creano continuamente, rivedono e provano le loro teorie sul mondo giocando con i loro giocattoli e amici. Secondo la teoria costruttivista dell’apprendimento di Piaget, i bambini sono costruttori attivi di conoscenza, non recipienti passivi. Ai bambini non *vengono* idee, i bambini *creano* idee.

Nei primi anni Sessanta, Seymour si trasferì da Ginevra, in Svizzera, a Cambridge, in Massachusetts, accettando un incarico come professore al MIT. Così facendo, Seymour si spostò dall'epicentro di una rivoluzione che riguardava lo sviluppo del bambino, all'epicentro di una rivoluzione tecnologica che riguardava i computer - e trascorse i successivi decenni a costruire connessioni tra le due rivoluzioni. Quando Seymour arrivò al MIT, i computer costavano ancora centinaia di migliaia di dollari, e venivano usati soltanto da grandi aziende, agenzie governative e università. Ma Seymour intuì che i computer sarebbero diventati accessibili a chiunque, perfino ai bambini, e formulò una visione di come il computer avrebbe potuto trasformare il modo in cui i bambini avrebbero imparato e giocato.

Seymour emerse ben presto come leader nella vivace battaglia intellettuale su come introdurre i computer nell'educazione. La maggior parte dei ricercatori ed educatori adottò un approccio di istruzione assistita dal computer, in cui i computer giocavano il ruolo degli insegnanti fornendo informazioni e istruzioni agli studenti, somministrando questionari per verificare cosa gli studenti avessero appreso, per poi adattare le istruzioni successive sulla base delle risposte degli studenti ai quiz.

Seymour aveva una visione radicalmente diversa. Per Seymour i computer non erano un sostituto degli insegnanti, bensì un nuovo mezzo di espressione, un nuovo strumento per costruire cose. Nel 1971, cinque anni prima che il primo personal computer fosse introdotto, Seymour fu coautore (con Cynthia Solomon) di un articolo intitolato: "Venti Cose da Fare con un Computer". L'articolo descriveva come i bambini avrebbero potuto usare il computer per disegnare immagini, creare giochi, controllare robot, comporre musica, e fare molte altre attività creative.

L'approccio di Seymour si basava su quello che aveva imparato da Piaget, una visione dei bambini come costruttori attivi, e non recipienti passivi, di conoscenza. Seymour fece un passo avanti, sostenendo che i bambini costruiscono la conoscenza più efficacemente quando sono attivamente coinvolti nel costruire qualcosa nel mondo — cioè quando sono costruttori di cose. Seymour chiamò il suo approccio costruzionismo, poiché mette insieme i due tipi di costruzioni: quando i bambini costruiscono qualcosa nel mondo, costruiscono nuove idee nella loro testa, il che li motiva a costruire nuove cose nella realtà e così via, in una spirale di apprendimento senza fine.

Per dare vita a queste idee, Seymour e i suoi colleghi svilupparono un linguaggio di programmazione per bambini chiamato Logo. Fino ad allora, programmare era considerata un'attività specialistica, accessibile soltanto a persone con competenze matematiche avanzate. Ma Seymour vide la programmazione come un linguaggio

universale per fare cose con il computer, e sostenne che tutti dovessero imparare a programmare.

Nel suo libro *Mindstorms*, Seymour mette in evidenza le differenze tra l'approccio di istruzione assistita dal computer, nel quale "il computer viene usato per programmare il bambino", con il suo approccio, nel quale "il bambino programma il computer". Nel processo di imparare a programmare, scrive, un bambino "acquisisce padronanza sulla più moderna e potente tecnologia stabilendo al tempo stesso un rapporto personale con alcune delle idee più profonde della scienza, della matematica, e dell'arte della costruzione intellettuale di modelli".

All'inizio, quando Logo fu sviluppato, i bambini lo usavano principalmente per controllare il movimento di una "tartaruga" robotica (chiamata così perché aveva un guscio per proteggere i componenti elettronici). Quando il personal computer divenne disponibile nella seconda metà degli anni '70, i bambini usarono Logo per disegnare immagini sullo schermo, digitando comandi come 'avanti 100' e 'destra 60' per dire alla "tartaruga sullo schermo" come muoversi, girare e disegnare. Mentre i bambini scrivevano programmi con Logo, imparavano concetti matematici in un modo significativo e motivante, in un contesto di lavoro su progetti che li appassionavano.

Durante gli anni '80, migliaia di scuole hanno insegnato a milioni di studenti a programmare in Logo, ma l'entusiasmo iniziale non è durato. Molti insegnanti e studenti avevano difficoltà a imparare a programmare con Logo, perché il linguaggio era pieno di sintassi non intuitiva e di punteggiatura. A peggiorare le cose, Logo era spesso introdotto con attività che non sostenevano l'interesse, né degli insegnanti né degli studenti. In molte classi Logo era insegnato come fine a se stesso, e non come un mezzo con cui gli studenti potessero esprimersi ed esplorare quelle che Seymour chiamava "idee potenti". Ben presto, molte scuole si spostarono su altri usi del computer. Iniziarono a vedere i computer come strumenti per distribuire e accedere a informazioni, piuttosto che per fare e creare come Seymour aveva immaginato.

Le idee di Seymour sull'apprendere costruendo stanno cominciando di nuovo a fare presa, come evidenziato dall'ascesa del Movimento Maker. Sebbene il lavoro di Seymour con Logo sia iniziato più di 50 anni fa e che il suo libro, pietra miliare, *Mindstorms* sia stato pubblicato nel 1980, le sue idee principali sono importanti e pertinenti, oggi più che mai.

## Fluency

Negli ultimi anni, c'è stata un'impennata di interesse nell'imparare a programmare il computer — o coding, come viene ora comunemente chiamato oggi. Ci sono migliaia di applicazioni, siti web, laboratori per aiutare i ragazzi a imparare a programmare. Il nostro software di programmazione Scratch è parte di questo trend — ma con una marcata differenza.

La maggior parte delle introduzioni al coding sono basate su *puzzle*. Ai ragazzi viene chiesto di creare un programma per far muovere un personaggio virtuale oltre alcuni ostacoli per raggiungere un obiettivo. Per esempio, muovere il robot di Star Wars BB-8 per raccogliere un frammento di metallo senza incappare nel bandito, o programmare R2-D2 per mandare un messaggio ai piloti ribelli. Quando i ragazzi creano programmi per risolvere questi puzzle, imparano le abilità e i concetti base dell'informatica.

Con Scratch, ci focalizziamo sui *progetti* invece che sui puzzle. Quando presentiamo Scratch ai ragazzi, li incoraggiamo a creare le loro storie interattive, giochi, e animazioni. I ragazzi partono con le loro idee e le trasformano in progetti che possono condividere con altre persone.

Perché concentrarsi sui progetti? Noi vediamo il coding come una forma di espressione e di padronanza di un linguaggio, proprio come la scrittura. Quando si impara a scrivere non è sufficiente imparare l'ortografia, la grammatica e la punteggiatura. È importante imparare a raccontare storie e a comunicare le proprie idee. Lo stesso vale per il coding. I puzzle possono andare bene per imparare la grammatica e la punteggiatura di base del coding, ma non aiutano a esprimere se stessi. Immaginate di provare a imparare a scrivere, semplicemente risolvendo le parole crociate. Potrebbe migliorare la vostra ortografia e il vostro vocabolario, e potrebbe essere divertente, ma potreste mai diventare dei bravi scrittori, in grado di raccontare storie ed esprimere le vostre idee con padronanza? Non credo. Un approccio incentrato sui progetti è il percorso migliore per padroneggiare sia la scrittura che il coding.

Anche se la maggior parte delle persone non diventeranno giornalisti o romanzieri di professione, imparare a scrivere è importante per tutti. Lo stesso vale per il coding — e per ragioni simili. La maggior parte delle persone non diventeranno programmatori o informatici, ma imparare a programmare fluentemente è prezioso per tutti. Avere padronanza sia con la scrittura che con il coding, aiuta a *sviluppare il proprio pensiero*, a *sviluppare la propria espressività* e a *sviluppare la propria identità*.

## **Sviluppare il proprio pensiero**

Nel processo di scrittura, si impara a organizzare, raffinare, e riflettere sulle proprie idee. Migliorando la capacità di scrittura si diventa migliori pensatori.

Anche quando si impara a programmare si diventa migliori pensatori. Ad esempio, si impara come suddividere problemi complessi in parti più semplici. Si impara a individuare problemi e a correggerli. Si impara a raffinare e a migliorare iterativamente i progetti nel tempo. L'informatica Jeannette Wing ha reso popolare il termine *pensiero computazionale* per riferirsi a questo tipo di strategie.

Una volta che si imparano queste strategie di pensiero computazionale, queste possono essere utili in tutti i tipi di attività di problem-solving e progettazione, non soltanto nel coding e nell'informatica. Imparando a correggere programmi informatici si diventa più preparati a capire che cosa non ha funzionato quando una ricetta di cucina va storta, o quando ci si perde seguendo le indicazioni di qualcuno.

Risolvere puzzle può essere utile a sviluppare alcune di queste abilità di pensiero computazionale, ma creare i propri progetti porta ancora più in là, aiutando a sviluppare la propria voce e la propria identità.

## **Sviluppare la propria voce**

Sia la scrittura che il coding sono forme di espressione, modi di comunicare le proprie idee agli altri. Quando si impara a scrivere, per esempio, si può mandare un messaggio di auguri di compleanno a un amico, inviare un articolo a un giornale locale o affidare le proprie emozioni a un diario.

Io vedo il codice come un'estensione della scrittura, che ci consente di “scrivere” nuovi tipi di cose — storie interattive, giochi, animazioni e simulazioni. Faccio un esempio dalla comunità online di Scratch. Qualche anno fa, il giorno prima della festa della mamma, decisi di usare Scratch per fare un biglietto di auguri interattivo per mia madre. Prima di iniziare, controllai se qualcun altro avesse fatto biglietti simili in Scratch. Digitai “Festa della Mamma” nella barra di ricerca, e fui deliziato dal vedere dozzine e dozzine di progetti — molti dei quali creati nelle 24 ore precedenti da procrastinatori come me!

Per esempio, uno dei progetti iniziava con le parole “HAPPY MOM DAY” (*N.d.T.*, “buona festa della mamma”) scritta su un grande cuore. Ognuna delle 11 lettere che componevano la frase era interattiva: quando veniva toccata dal cursore del mouse, la lettera si trasformava in una parola. Non appena mossi il cursore del mouse sullo schermo, toccando ogni lettera venne fuori un messaggio speciale dedicato alla festa della mamma: “Ti voglio bene e ci tengo a te. Buona festa della mamma”.

Il creatore di questo progetto aveva chiaramente sviluppato la propria voce con Scratch — imparando a esprimersi in modi nuovi e integrando il coding nella sua quotidianità. Credo che in futuro diventerà naturale per i giovani esprimersi attraverso il coding così come attraverso la scrittura.

(In ogni caso, finii per non fare il biglietto di auguri per la mia mamma. Le mandai invece dozzine di link ai progetti della festa della mamma che avevo trovato sul sito di Scratch. Mia mamma, un'educatrice per tutta la vita, rispose con questo messaggio: “Mitch, mi è piaciuto tantissimo vedere tutti i biglietti dei ragazzi di Scratch ... e adoro essere la madre di un figlio che ha aiutato a dare ai ragazzi uno strumento per celebrare in questo modo!!!!”)

### **Sviluppare la propria identità**

Quando le persone imparano a scrivere, cominciano a vedere se stesse in modo diverso - e a vedere il loro ruolo nella società in modo diverso. L'educatore e filosofo brasiliano Paulo Freire condusse una campagna di alfabetizzazione per le comunità povere non soltanto per aiutare le persone a trovare un lavoro, ma anche per aiutarle a imparare che “potevano fare e rifare se stesse” (come scrisse in *Pedagogia dell'Indignazione*).

Vedo lo stesso potenziale per il coding. Nella società odierna, le tecnologie digitali sono un simbolo di possibilità e progresso. Quando i bambini imparano a usare le tecnologie digitali per esprimersi e condividono le loro idee attraverso il coding, iniziano a vedere se stessi in modo nuovo. Cominciano a vedere la possibilità di contribuire attivamente alla società. Cominciano a vedersi parte del futuro.

Avendo offerto Scratch ai giovani, mi entusiasma quello che creano — e quello che imparano nel processo. Ma ciò che maggiormente mi rende entusiasta è la modalità in cui molti *Scratcher* iniziano a vedersi come creatori, sviluppando orgoglio e confidenza nelle proprie abilità di creare cose e di esprimersi fluentemente con le nuove tecnologie.