

CURRICOLO VERTICALE CODING E ROBOTICA EDUCATIVA

Introduzione

Il Curricolo è un percorso formativo, con traguardi da raggiungere strada facendo, che occorre adeguatamente pianificare. Si deve tener conto di obblighi dati dal raggiungimento di obiettivi e di strategie didattiche per l'apprendimento nelle varie età. La progressione didattica relativa al Coding, che qui definiamo nei vari livelli di passaggio da un ordine di scuola all'altro, va sviluppata in verticale dalla più tenera infanzia fino all'Università e anche oltre, in riferimento all'apprendimento lungo tutto l'arco della vita (lifelong learning).

Il pensiero computazionale come competenza trasversale

Il pensiero computazionale si configura oggi come **QUARTA ABILITÀ di base** dopo leggere, scrivere e fare di conto. Per questo dovrebbe essere insegnato fin dai primi anni di scuola

Coding è un termine inglese al quale corrisponde in italiano la parola *programmazione*, ma non nel senso più tradizionale dell'espressione. Con il coding bambini e ragazzi sviluppano il **pensiero computazionale**, l'attitudine a risolvere problemi più o meno complessi. Quando si parla di **pensiero computazionale**, si parla di un approccio inedito alla soluzione dei problemi e la palestra del pensiero computazionale è il **coding**, che consente di apprendere le basi della programmazione informatica in modo pratico e divertente. La scelta di un approccio ludico è strategica.

Il Coding rappresenta uno strumento **trasversale a tutte le discipline**, che favorisce lo sviluppo logico del pensiero, un approccio curioso di fronte alla realtà e la capacità di provare a **risolvere i problemi o di ripartire dagli errori** o dagli ostacoli incontrati nei processi formativi. Il coding diventa uno strumento che accompagna l'allievo nel maturare il proprio pensiero logico; un approccio metodologico che lo aiuta ad apprendere e a sviluppare le capacità di analisi, di scomposizione di un dato problema; gli permette di comprendere le ragioni di un determinato fattore e di poter elaborare delle soluzioni. Non si impara solo a programmare ma si programma per apprendere.

Il termine "**pensiero computazionale**" è entrato nella pratica didattica col Piano Nazionale Scuola Digitale (PNSD), che ha riconosciuto

attività di questo tipo come **essenziali per la formazione** degli studenti nell'era digitale.

Strumenti come Code.org, Scratch, CodyRoby, CodyFeet, ecc, possono essere utilizzati in vari percorsi formativi.

Un altro strumento molto efficace per lo sviluppo del pensiero computazionale è la **robotica educativa** come concretizzazione di quanto gli alunni “progettano” con il coding. Attraverso strumenti pensati appositamente per la didattica (*Bee Bot, cubetto, Doc, Mind, Lego Wedo, microbit, Arduino ecc*) si può completare il percorso formativo iniziato con il coding, utilizzando gli algoritmi per “programmare” robot o altre entità tecnologiche.

Obiettivi generali del percorso

- Introdurre il modello del “Pensiero Computazionale” come metodo per la risoluzione dei problemi che aiuta a riformularli per poterli risolvere efficientemente in modo automatico.
- Introdurre in modo intuitivo e ludico i concetti base della programmazione per sviluppare il pensiero computazionale.
- Analizzare e organizzare i dati del problema in base a criteri logici.
- Rappresentare i dati del problema tramite opportune astrazioni.
- Formulare il problema in un formato che ci permette di usare un “sistema di calcolo” (nel senso più ampio del termine, ovvero una macchina, un essere umano, o una rete di umani e macchine) per risolverlo.
- Automatizzare la risoluzione del problema definendo una soluzione algoritmica, consistente in una sequenza accuratamente descritta di passi, ognuno dei quali appartenente ad un catalogo ben definito di operazioni di base.
- Identificare, analizzare, implementare e verificare le possibili soluzioni con un’efficace ed efficiente combinazione di passi e risorse (avendo come obiettivo la ricerca della soluzione migliore secondo tali criteri).
- Generalizzare il processo di risoluzione del problema per poterlo trasferire ad un ampio spettro di altri problemi.
- Offrire esperienze di progettazione robotica per sviluppare capacità di problem solving ed estese abilità tecnologiche.

Scuola dell'infanzia

Nella scuola dell'infanzia, si tratta di iniziare con i bambini un percorso per sviluppare il loro pensiero computazionale in un contesto di gioco, affinché comincino a "ragionare" secondo i nuovi linguaggi informatici.

La scuola dell'infanzia diventa una palestra dove i bambini allenano il loro pensiero computazionale, ma dove allo stesso tempo si cerca di guidarli all'uso attivo e consapevole delle tecnologie, offrendo loro la possibilità di sperimentare nuove modalità e nuovi contesti per riflettere, cooperare, sviluppare la creatività e imparare; ma anche l'opportunità di essere dalla parte di chi programma per creare e non solo di chi semplicemente clicca per usufruire.

La costruzione di queste competenze può contribuire a favorire lo sviluppo di elementi di progettualità anche in ambiti disciplinari diversi da quello informatico. Coding e pensiero computazionale rendono possibile apprendere strategie di risoluzione di problemi, progettazione e comunicazione anche a chi non programmerà mai.

Grazie alle attività che si possono svolgere sia in un ambiente tecnologico, con l'utilizzo del computer e della rete internet, ma soprattutto in modalità "unplugged", non si rende necessario scrivere i comandi, le "stringhe" di codice, ma si impara a mettere in sequenza logica le raffigurazioni grafiche che li rappresentano ed è possibile acquisire alcune capacità come:

- capire cos'è un algoritmo: facendo scoprire ai bambini che sono algoritmi alcuni dei modi di operare, nella vita di tutti i giorni o a scuola, che realizziamo (quasi) automaticamente;
- usare il ragionamento logico per spiegare il funzionamento di alcuni semplici algoritmi;
- capire i principi alla base del funzionamento di un computer;
- selezionare, trasportare e lasciare: attività di drag and drop.

SCUOLA DELL'INFANZIA		
Periodo	Metodologia	Verifiche
Secondo e terzo anno della scuola dell'infanzia	<ul style="list-style-type: none"> - Favorire la curiosità, la scoperta, l'esplorazione concreta, il gioco, il procedere per tentativi, la collaborazione, la riflessione sulle esperienze; - Learning by doing; - Cooperative Learning; - Tutoring 	<ul style="list-style-type: none"> - Osservazione - Prodotti finali realizzati (attività unplugged, animazioni, simulazioni, storytelling, disegni rappresentativi delle attività svolte).
Raccordi tra i campi di esperienza <ul style="list-style-type: none"> - Capacità di comunicare l'esperienza vissuta attraverso la drammatizzazione, la narrazione e il disegno. - Sviluppare capacità di problem solving. - Collaborare e interagire con gli altri per giungere alla soluzione di un problema - Esplorare varie forme di narrazione digitale e animazioni. 		
Competenze	Abilità	Conoscenze
<ul style="list-style-type: none"> - Giocare in modo costruttivo e creativo con gli altri, argomentare, confrontarsi, sostenere le proprie ragioni con adulti e bambini. - Provare piacere nel movimento e sperimentare schemi posturali e motori, applicarli nei giochi individuali e di gruppo, anche con l'uso di piccoli attrezzi adattandoli a situazioni ambientali all'interno della scuola e all'aperto. - Utilizzare materiali e strumenti, tecniche espressive e creative; 	<ul style="list-style-type: none"> - Formulare tanti "perché" sulle questioni concrete - Riflettere sul senso e le conseguenze delle azioni - Esprimersi e comunicare attraverso il corpo - Affinare le capacità percettive e di conoscenza degli oggetti - Orientarsi nello spazio - Cercare, scoprire, giocare, saltare, correre - Familiarizzare con l'esperienza della multimedialità (la fotografia, il cinema, la televisione, il digitale) e ricercare le loro possibilità espressive e creative. 	<ul style="list-style-type: none"> - Giocare esplorando l'ambiente - Operare e giocare con materiali strutturati, costruzioni e giochi da tavolo di vario tipo - Ascoltare narrazioni - Attività di coding unplugged: muoversi su una scacchiera come un robot, eseguendo semplici istruzioni di movimento impartite dai compagni - CodyFeet - Cody&Roby

<ul style="list-style-type: none"> - esplorare le potenzialità offerte dalle tecnologie. - Esplorare e sperimentare le prime forme di comunicazione attraverso la scrittura, incontrando anche le tecnologie digitali e i media. - Elaborare procedimenti creativi per la risoluzione di problemi - Sviluppare l'interesse per macchine e strumenti tecnologici, scoprire le funzioni e i possibili usi. - Esplorare e sperimentare le possibilità offerte dalle tecnologie per fruire delle diverse forme artistiche, per comunicare e per esprimersi attraverso di esse. 	<ul style="list-style-type: none"> - Chiedere spiegazioni - Confrontare punti di vista - Progettare giochi e attività - Elaborare e condividere conoscenze - Estendere il lessico - Fare esperienza di lavoro di gruppo - Toccare, smontare, costruire e ricostruire, affinando i gesti - Individuare qualità e proprietà degli oggetti e dei materiali - Imparare a rappresentare con simboli semplici i risultati delle loro esperienze - Scegliere ed eseguire i percorsi più idonei per raggiungere una meta prefissata 	<ul style="list-style-type: none"> - Giochi di coding utilizzando un robot programmabile - Uso di strumenti informatici - Utilizzare programmi specifici per l'infanzia, al fine di familiarizzare con le esperienze della multimedialità. - Altri strumenti di comunicazione e loro uso (audiovisivi, tablet, ecc). - Alla scoperta dei PIXEL
<p style="text-align: center;">Competenze in uscita</p> <ul style="list-style-type: none"> - Conoscere i blocchi "Vai avanti", "Gira a destra"; "Gira a sinistra"; Ripetizioni. - Usare il ragionamento logico per comprendere cosa sono gli oggetti smart e comprendere il funzionamento di alcuni semplici algoritmi; - Attività unplugged: orientarsi sul tappeto a scacchiera, CodyRoby - Primo approccio all'ambiente editor Scratch junior, Storytelling. 		

Scuola Primaria

Il Coding è una prima forma di approccio interdisciplinare alle TIC: consente l'avvio all'uso consapevole del computer per comprendere che le dotazioni tecnologiche sono strumenti attraverso i quali realizzare dei progetti; sviluppa il pensiero riflessivo e procedurale; stimola la riflessione sull'errore come nuovo spunto di lavoro; sviluppa le capacità di analisi sul proprio operato; incrementa le capacità di espressione linguistica sia orale che scritta per comunicare il proprio operato agli altri o come memoria personale; garantisce un utilizzo diretto di conoscenze matematiche, linguistiche, antropologiche e scientifiche per sostanziare di contenuti gli elaborati prodotti; sviluppa il lavoro cooperativo e delle abilità individuali.

Nella scuola primaria l'obiettivo è sviluppare tutte queste competenze e capacità attraverso lo studio dei principali costrutti del coding (iterazione, istruzioni condizionali, funzioni).

DALLA SCUOLA DELL'INFANZIA ALLA SCUOLA PRIMARIA	
Competenze di base	Conoscenze
<ul style="list-style-type: none">- Avere consapevolezza delle proprie risorse e dei propri limiti- Saper chiedere aiuto quando occorre- Manifestare curiosità e voglia di fare- Condividere esperienze e giochi- Cogliere diversi punti di vista- Utilizzare gli errori come fonte di conoscenza- Saper raccontare, narrare, descrivere situazioni ed esperienze vissute- Sapersi orientare nel mondo dei media, delle tecnologie- Formulare ipotesi, ricercare soluzioni a situazioni problematiche- Prestare attenzione alle consegne	<ul style="list-style-type: none">- Attività di coding unplugged: muoversi su una scacchiera come un robot, eseguendo semplici istruzioni di movimento impartite dai compagni- CodyFeet (primo periodo classe prima)- Cody & Roby- Giochi di coding utilizzando robot programmabili- Giochi al computer- Alla scoperta dei Pixel

SCUOLA PRIMARIA

Periodo	Metodologia	Verifiche
Intero corso di studi (5 anni, dalla prima alla quinta classe di scuola primaria).	<ul style="list-style-type: none"> - Attività di approccio mediato dal docente, - Learning by doing; Cooperative Learning; - tutoring; didattica dell'errore (debug). 	<ul style="list-style-type: none"> - Osservazione - prodotti finali realizzati (attività unplugged, animazioni, simulazioni, videogiochi).
<p style="text-align: center;">Raccordo con le discipline scolastiche (interdisciplinarietà)</p> <ul style="list-style-type: none"> - Capacità di comunicazione orale e scritta, - applicare le conoscenze curricolari apprese nelle varie classi negli ambiti linguistici, matematici, scientifici, tecnologici, antropologici, etc, per la pianificazione ed elaborazione dei prodotti - Sviluppare capacità di problem solving - Collaborare e interagire con gli altri per giungere alla soluzione di un problema - esplorare varie forme di narrazione digitale, animazioni e creazioni di videogiochi. 		
Competenze	Abilità	Conoscenze
<ul style="list-style-type: none"> - Sapersi esprimere e comunicare utilizzando codici e linguaggi diversi - Utilizzare con consapevolezza i concetti di ripetizione e condizione - Analizzare e rappresentare processi utilizzando modelli logici - Approcciarsi ad un uso consapevole delle TIC - Utilizzare trasversalmente le conoscenze 	<p>COSTRUIRE ABILITÀ DI BASE</p> <ul style="list-style-type: none"> - Indicare la destra e la sinistra rispetto al personaggio da muovere sullo schermo - Imparare a scrivere i comandi in ordine e risolvere i problemi utilizzando una serie di istruzioni. - Utilizzare i blocchi di ripetizione. - Riconoscere in una dichiarazione logica sia l'ipotesi sia la conclusione. - Utilizzare le espressioni condizionali se/allora - Applicare il concetto di funzione nella soluzione di schemi logici <p>LEGGERE E SCRIVERE IL CODICE</p> <ul style="list-style-type: none"> - Identificare istruzioni condizionali. - Creare istruzioni condizionali. 	<ul style="list-style-type: none"> - Imparare cos'è un ciclo, quando usarlo. - Conoscere le espressioni condizionali se/allora - Conoscere il concetto logico di funzione - Conoscere il costrutto di ripetizione - conoscere le parti di un device - Conoscere l'ambiente di programmazione visuale Scratch jr (classe prima) e di Scratch (dalla seconda) - Conoscere un editor grafico (come ad es. quello per editare gli Sprite in Scratch) - Conoscere le modalità di programmazione dei robot

	<ul style="list-style-type: none"> - Utilizzare espressioni condizionali, cicli, funzione, ripetizione - Scrivere codice conciso - Apportare modifiche al codice - Identificare gli errori nel codice e dedurre soluzioni corrette. - Progettare e costruire storie tecnologiche usando le potenzialità di Scratch - Stabilire relazioni causa-effetto - Creare semplici programmi per istruire i robot <p>PROBLEM SOLVING</p> <ul style="list-style-type: none"> - Logicamente applicare la soluzione più efficace a problemi che possono essere risolti in molteplici modi. - Saper esprimere i propri punti di vista e le proprie proposte - Saper aprire un programma, utilizzarlo, salvare il lavoro, riaprirlo - Comprendere che la macchina è un mezzo e non un fine del lavoro - Utilizzare le conoscenze disciplinari per riempire di contenuti le attività 	
<p style="text-align: center;">CONTENUTI</p> <ul style="list-style-type: none"> - Concetto di programmazione (programmatore, linguaggio di programmazione, sequenza) - Il linguaggio delle cose: oggetti "smart" e non - L'ora del codice corso rapido 20 ore - Code.org Corso 1 e 2, per le prime tre classi della primaria - Code.org Corso 3 per le ultime classi della primaria - Code.org: costruire un gioco, una storia. - Eu code week - Code days - Scratch Jr, per la prima classe della primaria - Scratch: inizializzazione di un progetto, Sprite, stage, costumi e sfondi; - modifica Sprite con editor grafico 		

- gioco, storytelling, grafica computerizzata: dialoghi e interazione tra personaggi per le animazioni;
- Sequenza e Algoritmi, espressioni condizionali, cicli. La funzione. La ripetizione
- creazione dei propri personaggi con editor grafico;
- uso della interattività e del movimento per la realizzazione di giochi multimediali;
- modalità per creare disegni geometrici con il codice, uso del suono.
- Approccio alla robotica: Doc, per le prime due classi della primaria, Mind e microbit, per le ultime classi;
- Altre esperienze di programmazione orientata sugli oggetti
- Pixel Art

Competenze in uscita

- Conoscenza dei blocchi "Vai avanti", "Gira a destra"; "Gira a sinistra"; "Ripeti"; "Se – Allora –Altrimenti"
- Conoscenza della piattaforma "L'Ora del Codice" di Programma il Futuro e di Code.org Corso 2 e 3
- (Ambiente editor Scratch Jr)
- Ambiente sviluppo Scratch: Sprite, stage, costumi e script; modifica Sprite con editor grafico inizializzazione di un progetto
- Primo conoscenza della robotica educativa e della programmazione di robot

Scuola Secondaria di Primo Grado

Nella scuola secondaria di primo grado l'obiettivo è sviluppare pensiero computazionale, attraverso l'utilizzo di costrutti più complessi della programmazione, per la realizzazione di algoritmi ottimizzati ed efficaci alla risoluzione di problemi, utilizzando la **anche** robotica educativa.

Si considerano i concetti base dell'informatica, adatti allo specifico livello di maturazione degli studenti, in modo indipendente dalla tecnologia.

L'utilizzo di strumenti didattici a difficoltà progressiva, come ad esempio il portale code.org, è indispensabile ad introdurre alcuni concetti piuttosto astratti, come ad esempio le funzioni con i parametri.

La prosecuzione delle attività con ambienti di programmazione visuali quali, per esempio, Scratch consente agli alunni di ideare e realizzare qualcosa di proprio: uno storytelling, un videogioco...

L'utilizzo, infine, di supporti legati alla robotica educativa consente di ridurre il livello di astrazione e di applicare gli algoritmi appresi per realizzare la programmazione di oggetti che interagiscono con il mondo reale (intelligenza delle cose).

Alcuni dei contenuti sottoelencati e dei programmi citati, che richiedono conoscenze più approfondite e capacità più avanzate, verranno trattati solo nel laboratorio opzionale di robotica educativa, nel quale gli studenti avranno modo di sviluppare maggiormente le abilità legate alla programmazione.

SCUOLA SECONDARIA DI PRIMO GRADO		
Periodo	Metodologia	Verifiche
L'intero triennio	<ul style="list-style-type: none">- Learning by doing;- Cooperative Learning;- tutoring;- didattica dell'errore (debug)	<ul style="list-style-type: none">- Osservazione- prodotti finali realizzati (attività unplugged, animazioni, simulazioni, videogiochi).

Raccordi con le discipline scolastiche (interdisciplinarietà)

- Matematica e Geometria, Tecnologia (disegno tecnico)
- Problem solving, logica e capacità di trasformare le proprie idee in App e Videogiochi come focus della programmazione
- Italiano e Lingue straniere
- Realizzazione di Storytelling con ambienti di sviluppo quali Scratch, realizzazione di app in lingua straniera
- Musica
- Realizzazione di programmi multimediali

Competenze	Abilità	Conoscenze
<ul style="list-style-type: none"> - Conoscere il concetto di algoritmo nelle sue varie forme ed applicazioni - Approfondire i concetti di variabile e di funzione con parametri - Approcciarsi ad una scrittura ottimizzata del codice - Conoscere le caratteristiche di un robot - Utilizzare trasversalmente le conoscenze 	<p>COSTRUIRE ABILITÀ DI BASE</p> <ul style="list-style-type: none"> - Imparare a scrivere algoritmi in maniera corretta e a risolvere problemi complessi scomponendoli in problemi più semplici. - Applicare il concetto di variabile nella soluzione di schemi logici - Utilizzare correttamente le funzioni con e senza parametri <p>LEGGERE E SCRIVERE IL CODICE</p> <ul style="list-style-type: none"> - Debugging - Scrivere codice ottimizzato - Apportare modifiche al codice in modo da ottimizzare lo stesso pur conservandone la funzionalità - Identificare i bug nel codice - Progettare e costruire storie tecnologiche unendole con il linguaggio di programmazione Scratch. - Remixare programmi esistenti <p>COSTRUIRE E PROGRAMMARE UN ROBOT</p> <ul style="list-style-type: none"> - Capire cosa sono e come usare sensori e motori per rendere interattivi i modelli - Stabilire relazioni causa-effetto - Creare semplici programmi per istruire i modelli LEGO <p>PROBLEM SOLVING</p> <ul style="list-style-type: none"> - Saper lavorare in gruppo. - Ottimizzare la soluzione di un problema 	<ul style="list-style-type: none"> - Conoscere il concetto logico di variabile - Conoscere il corretto utilizzo delle funzioni nella realizzazione di un algoritmo - Conoscere il concetto di parametro ed il suo utilizzo all'interno delle funzioni - Conoscere le principali funzioni di un programma ed utilizzarle in modo congruo - Conoscere l'ambiente di programmazione Scratch - Conoscere l'ambiente di sviluppo App Inventor - Conoscere le caratteristiche di un robot (sensori e motori) - Conoscere l'ambiente di progettazione Lego Mindstorm - Conoscere l'utilizzo di robot mBot e del suo ambiente di programmazione - Conoscere le basi di utilizzo della scheda Arduino

	<ul style="list-style-type: none"> - Scomporre un problema complesso in problemi di complessità inferiore - Utilizzare registri linguistici adatti agli interlocutori - Saper realizzare un algoritmo utile per la soluzione di un problema generico - Utilizzare le conoscenze disciplinari per riempire di contenuti le attività: - realizzare storytelling; - conoscere le principali regole nella realizzazione di figure geometriche anche di complessità elevata (es. circonferenza); - realizzare programmi di musica ecc 	
--	---	--

CONTENUTI

- L'ora del codice: Minecraft, Frozen, Ballando con il codice
- Il linguaggio delle cose: inventiamo oggetti "smart"
- Code.org corso rapido 20 ore
- Eu code week
- Approfondimento del concetto di Debug
- Creazione di un videogioco con Code.org
- Code days
- Remix di semplici progetti nati con Scratch
- Ambienti di sviluppo Scratch: Pixel e coordinate dello schermo
- Aspetto e utilizzo dei costumi
- Controllo
- Movimento
- Sensori (attendi, sta toccando) Stage ed utilizzo degli sfondi Utilizzo di Variabili e Funzioni
- Messaggi (comunicazione tra gli Sprite)
- Creazione di uno Storytelling con Scratch
- Creazione di un videogioco con Scratch
- Introduzione alla programmazione ad oggetti
- Oggetti e loro metodi
- Sensori e Timer
- Utilizzo software di progettazione Lego Mindstorm (costruire un robot)
- Utilizzo software di programmazione Lego Mindstorm (programmare un robot):
- Realizzazione di un proprio robot Lego Mindstorm da far interagire con altri robot simili
- Utilizzo di robot mBot che si basano su schede Arduino e ambienti di programmazione simile a Scratch
- Costrutti principali utilizzati nella programmazione dei robot

(Appendice del curriculum verticale di Istituto)