



Autonomia n. 86

ISTITUTO COMPRENSIVO STATALE

“Rubino NICODEMI”

VIA ROMA,47 TEL. E FAX 089891238 – 84084 FISCIANO

C.F. 80042300659 - C.M. SAIC89000V

e-mail : [saic89000v@istruzione.it](mailto:saic89000v@istruzione.it) - [saic89000v@pec.istruzione.it](mailto:saic89000v@pec.istruzione.it)

## Progetto “Coding in Rubino, now!”

### Curricolo verticale

#### 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 sta a noi qui definire nei vari livelli di passaggio da un ordine di scuola all'altro, va sviluppata in verticale dalla più tenera infanzia fino all'Università e ben oltre, in riferimento all'apprendimento lungo tutto l'arco della vita (*lifelong learning*). La formazione protratta lungo tutto l'arco della vita attiva è alla base dell'effettiva crescita della persona, in ogni ambito della vita professionale e sociale.

Nella educazione formale (quella erogata dai sistemi scolastici) le competenze digitali non costituiscono una materia a se stante, bensì un insieme di attività trasversali che interessano e pervadono tutte le discipline insegnate. Anche la diffusione del Coding si espande in un compendio di progettazione e pianificazione trans-curricolare che investe ogni disciplina.

**Coding** ovvero il gerundio o participio presente del verbo to code: codificando o codificante.

Codificante di che? In questa nostra verva nella anglicizzazione dei termini ci dimentichiamo spesso che l'italiano ha bisogno di essere più preciso e meno volatile dell'inglese. Andrebbe ricordato che codificare significa molte cose e non solo programmare i computer.

Programmare i computer (anche detti in italiano elaboratori elettronici) forse ci sembrava troppo scontato o difficile ed infatti dando una occhiata a wikipedia:

*La programmazione, in informatica, è l'insieme delle attività e tecniche che una o più persone specializzate, programmatori o sviluppatori (developer), svolgono per creare un programma, ossia un software da far eseguire ad un computer, scrivendo il relativo codice sorgente in un certo linguaggio di programmazione.*

Per gli anglosassoni invece:

*Computer programming (often shortened to programming) is a process that leads from an original formulation of a computing problem to executable computer programs. Programming involves activities such as analysis, developing understanding, generating algorithms, verification of requirements of algorithms including their correctness and resources consumption, and implementation (commonly referred to as coding) of algorithms in a target programming language. Source code is written in one or more*

*programming languages. The purpose of programming is to find a sequence of instructions that will automate performing a specific task or solving a given problem. The process of programming thus often requires expertise in many different subjects, including knowledge of the application domain, specialized algorithms and formal logic.*

Le definizioni sono notevolmente diverse e rispecchiano inevitabilmente il modo di pensare dei due popoli, perché anche la lingua che è espressione di un popolo forma il suo modo di pensare. Per l'italiano programmare è una attività tecnica svolta da persone specializzate mentre per gli anglosassoni è un processo cognitivo.

## **Premessa**

Il Curricolo costituisce la descrizione di un percorso che si vuole costruire e delle mete che si intendono raggiungere, data una situazione di partenza. Sottesa all'idea di curricolo verticale vi è quella di continuità che a sua volta richiama il concetto di competenza. Essere competenti significa saper trasferire in contesti di vita reali, dimostrando autonomia e responsabilità, conoscenze e abilità acquisite in contesti educativi formali. Attraverso il curricolo verticale si intende, dunque, perseguire la destinazione di scopo del nostro sistema d'istruzione che si sostanzia nella realizzazione del successo formativo di tutti e di ciascuno.

Nella stesura di un curricolo verticale di coding non si può prescindere dai documenti europei, in primis dalla Raccomandazione del Parlamento europeo del 2006 che fra le otto competenze chiave per l'apprendimento permanente annovera la competenza digitale.

Essere competenti dal punto di vista digitale non significa solo saper utilizzare oggetti digitali o saper ricavare l'informazione dai social media, ma significa anche conoscere la logica e il funzionamento dei media digitali per favorirne un uso consapevole e responsabile. Il fine ultimo è quello di formare nelle aule utenti attivi in grado di decidere autonomamente cosa consumare e come consumarlo. Questa è, peraltro, una delle nuove alfabetizzazioni previste dal documento "La Buona scuola", infatti tra le linee guida del progetto è citata anche l'«educazione al pensiero computazionale e al coding nella scuola italiana».

Il concetto chiave è il «pensiero computazionale», che ricorre anche nel documento del governo.

Il pensiero computazionale è la capacità di individuare un procedimento costruttivo, fatto di passi semplici e non ambigui, che porta alla soluzione di un problema complesso. Non riguarda tanto un problem solving da affrontare in prima persona, ma conduce a specificare procedure e istruzioni che un "esecutore" può eseguire autonomamente; significa pensare in maniera algoritmica ovvero trovare una soluzione e svilupparla.

Il coding dà ai bambini una forma mentis che permetterà loro di affrontare problemi complessi quando saranno più grandi. Insomma imparare a programmare apre la mente.

## Programma

Nel mondo attuale, dove ci si collega a internet da ovunque vogliamo e con diversi strumenti, è importante la cultura della fruizione non passiva; in questo ci viene in aiuto il pensiero computazionale.

Il pensiero computazionale è il riuscire a mettere in pratica, in una serie di azioni (singole, precise e consequenziali), un'idea che abbiamo avuto per risolvere un determinato pensiero. Attraverso il *coding* si cerca di trasmettere ai cosiddetti “nativi digitali” sia il linguaggio della programmazione e degli algoritmi (mettendo in primo piano la logica che sta dietro alla tecnologia) sia la potenza della segmentazione di un problema e il loro affidamento a piccoli gruppi. Così si dà forza al lavoro collaborativo, al rispetto delle diverse idee.

Le "Indicazioni nazionali per il curricolo della scuola dell'Infanzia e del Primo ciclo dell'Istruzione (2012)" fanno riferimento all'informatica e alla programmazione senza però precisare bene che cosa si intenda e non fanno nessun riferimento al *Coding* e al pensiero Computazionale. Dato che il quadro di riferimento delle Indicazioni sono le otto competenze-chiave per l'apprendimento permanente definite dal Parlamento europeo e dal Consiglio dell'Unione Europea, possiamo prendere in considerazione la competenza chiave nel campo digitale che “consiste nel saper utilizzare con dimestichezza e spirito critico le tecnologie della società dell'informazione per il lavoro, il tempo libero e la comunicazione”. E, quindi, perché non inserire in tale ambito il pensiero computazionale che costituisce il contesto scientifico che fonda e rende possibili, per esempio, le attività di progettazione, sviluppo e funzionamento del web e delle app?

La costruzione di questa 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 e non costruirà mai nuove app.

Le **attività** proposte saranno sia di tipo tradizionale (con uso di carta e penna) sia di tipo tecnologico sfruttando la piattaforma ministeriale “Programma il futuro” e Code.org. Attraverso attività (costruzione di labirinti, percorsi, disegni...) essenzialmente giocose, gli alunni si avvicineranno al linguaggio della programmazione in modo naturale e inizieranno ad affrontare i problemi scomponendoli in parti sempre più piccole, ponendo l'attenzione sul processo logico, seguendo procedure (ALGORITMI) creati da loro. **Possiamo applicare procedure a qualsiasi disciplina ecco perché il CODING è una metodologia trasversale a tutte le discipline.**

Grazie alle attività che si possono svolgere nel progetto “Programma Il Futuro”, consistenti in esercizi “guidati” sul pensiero computazionale (svolti sia in un ambiente tecnologico, con l'utilizzo del computer e della rete internet sia in modalità “unplugged”), alle attività proposte in Code Week e la conoscenza di altri ambienti di programmazione visuale 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;
- capire i principi alla base del funzionamento di Internet;
- usare la tecnologia digitale in modo sicuro, rispettoso e responsabile;
- selezionare, trasportare e lasciare: attività di *drag and drop*.

# Programma

<b>FINALITÀ'</b>	Portare il coding ad essere quarta competenza di base, insieme al leggere, scrivere e far di conto.
<b>OBIETTIVI E RISULTATI ATTESI</b>	<ul style="list-style-type: none"><li>• Introdurre in modo intuitivo e ludico i concetti base della programmazione per sviluppare il pensiero computazionale.</li><li>• Introdurre il modello del “Pensiero Computazionale” come metodo per la risoluzione di problemi che aiuta a riformularli per poterli risolvere efficacemente in modo automatico</li><li>• Essere in grado di “codificare” i singoli passi da fare per risolvere un problema.</li><li>• Saper applicare le modalità operative del coding in maniera trasversale a tutte le discipline o alle situazioni problematiche della vita quotidiana.</li><li>• Imparare a lavorare in gruppo, a programmare le attività e soprattutto a “prevedere il futuro”, divertendosi.</li></ul>
<b>COMPETENZE PROMOSSE</b>	<ul style="list-style-type: none"><li>• Comunicazione nella madrelingua</li><li>• Competenza digitale: Utilizzare con dimestichezza e spirito critico le tecnologie della società dell’informazione per il lavoro, il tempo libero e la comunicazione.</li><li>• Competenza matematica: Utilizzare le strategie del pensiero razionale negli aspetti dialettici e algoritmici per affrontare situazioni problematiche, elaborando opportune soluzioni.</li><li>• Imparare ad imparare: partecipare attivamente alle attività portando il proprio contributo personale.</li><li>• Competenze sociali e civiche: collaborare e partecipare comprendendo i diversi punti di vista delle persone.</li><li>• Spirito d’iniziativa e imprenditorialità: risolvere i problemi che si incontrano nella vita e nel lavoro e proporre soluzioni; scegliere tra opzioni diverse; progettare e pianificare.</li></ul>
<b>SVOLGIMENTO</b>	<p>Le attività si svolgeranno in classe, senza un orario predeterminato, sviluppandosi in maniera trasversale laddove le diverse attività disciplinari normalmente programmate richiederanno un ragionamento analitico e l’individuazione dei diversi passaggi necessari per sviluppare un argomento o un problema.</p> <p>La classe sarà attivamente coinvolta in manifestazioni a carattere nazionale o internazionale, con attività unplugged o tecnologiche, producendo anche materiali, video, documentazione, manufatti che mettano in pratica i principi acquisiti.</p>
<b>ESPERTI INTERNI</b>	Gli insegnanti di classe
<b>ESPERTI ESTERNI</b>	Ogni qualvolta sarà possibile la classe, con l’ausilio della LIM, assisterà attivamente in streaming a lezioni on line tenute dal Prof Alessandro Bogliolo dell’Università di Urbino o da suoi collaboratori o docenti facenti parte della comunità Coding in Your classroom now.
<b>METODOLOGIE</b>	<ul style="list-style-type: none"><li>• Cooperative learning</li><li>• Learning by doing</li><li>• Debugging (didattica dell’errore)</li><li>• Attività laboratoriali di gruppo, unplugged (senza computer e senza rete) e tecnologiche, basate sul gioco e su una didattica</li></ul>

	<p>della scoperta.</p> <ul style="list-style-type: none"> <li>• Brain storming</li> <li>• Posing e Problem solving</li> <li>• Attività di approccio mediato dal docente</li> <li>• Conversazioni collettive e tutoring</li> </ul>
STRUMENTI	<ul style="list-style-type: none"> <li>• <b>CodyRoby</b> è un metodo di programmazione unplugged fai da te basato su semplici carte da gioco che possono essere utilizzate per creare giochi da tavolo per ogni età o attività motorie.</li> <li>• <b>CodyWay</b> è un metodo di programmazione unplugged fai da te che consente di usare i percorsi nel mondo reale per fare esperienze di programmazione.</li> <li>• <b>Code.org</b> è un'organizzazione non-profit che promuove la diffusione del pensiero computazionale proponendo l'ora di codice (Hour of code) e offrendo strumenti didattici online ludici e intuitivi per giocare con la programmazione. In Italia il metodo di Code.org è adottato da Programma il futuro, l'iniziativa del Consorzio CINI e del MIUR.</li> <li>• <b>Scratch</b> è un linguaggio di programmazione visuale sviluppato al MIT Media Lab e reso disponibile online e offline attraverso una piattaforma che consente a chiunque di creare e condividere veri e propri programmi.</li> <li>• <b>“Programma il futuro”</b> parte da un'esperienza di successo avviata negli USA, che ha visto nel 2013 la partecipazione di circa 40 milioni di studenti e insegnanti di tutto il mondo. L'Italia è uno dei primi Paesi al mondo a sperimentare l'introduzione strutturale nelle scuole dei concetti di base dell'informatica attraverso la programmazione (coding ), usando strumenti di facile utilizzo che non richiedono un'abilità avanzata nell'uso del computer.</li> </ul>

Scuola dell'infanzia				
Anno	Competenze	Contenuti	Strumenti	Motivazione
Dai 4 ai 5 anni	Giocare con gli strumenti tecnologici e non, con un approccio aperto alla curiosità e all'esplorazione	<p>Giochi di esplorazione dell'ambiente</p> <p>Giochi di movimento su grandi scacchiere</p> <p>Muovere giocattoli /oggetti sulle scacchiere</p> <p>Gli strumenti digitali (pc, tablet, Lim, Smart Toys)</p>	<p>Scacchiere di grandi dimensioni da stendere sul pavimento;</p> <p>Scacchiere più piccole;</p> <p>Cody &amp; Roby;</p> <p>Bee Bot</p> <p>Blue Bot</p> <p>I-Pad</p> <p>Tablet</p>	<p>La curiosità, l'azione diretta, il gioco.</p> <p>Utilizzare le TIC come strumento di sviluppo dell'apprendimento, per permettere ai bambini di imparare attraverso una pluralità di canali.</p>

## Scuola Primaria

Periodo	Strumenti	Controllo Degli Apprendimenti
intero corso di studi	Code.org Scratch CodyRoby, Codyway Programma il futuro	Gli stessi prodotti realizzati

### Raccordi con la Scuola dell'Infanzia

- Conoscenza dei blocchi “Vai avanti”, “Gira a destra”; “Gira a sinistra”;
- Usare il ragionamento logico per spiegare il funzionamento di alcuni semplici algoritmi;
- Attività unplugged con CodyRoby e Codyway
- L’Ora del Codice di Programma il Futuro;
- Code org. Corsi 1, 2, 3 e 4

### Raccordi con la Scuola Secondaria di I Grado

- Conoscenza dei blocchi “Vai avanti”, “Gira a destra”; “Gira a sinistra”;
- L’Ora del Codice di Programma il Futuro;
- Corso Rapido di 20 ore di Code.org
- Ambienti editor Scratch: sprite, stage, costume e script

Competenze	Abilità	Conoscenze
<p>Sapersi esprimere e comunicare utilizzando codici e linguaggi diversi;</p> <p>L’alunno utilizza con consapevolezza i concetti di funzione, di variabile e di ripetizione</p> <p>Analizzare e rappresentare processi utilizzando modelli logici</p> <p>Approcciarsi ad un uso consapevole delle TIC</p> <p>Utilizzare trasversalmente le conoscenze</p>	<p style="text-align: center;"><b>COSTRUIRE ABILITA’ DI BASE</b></p> <ul style="list-style-type: none"> <li>- Indicare la destra e la sinistra rispetto al personaggio da muovere sullo schermo;</li> <li>- Imparare a scrivere i comandi in ordine e risolvere i problemi utilizzando algoritmi;</li> <li>- Utilizzare il blocco di ripetizione;</li> <li>- Riconoscere in una dichiarazione logica sia l’ipotesi che la conclusione;</li> <li>- Utilizzare le espressioni condizionali se/allora;</li> <li>- Applicare il concetto di funzione nella soluzione di schemi logici;</li> <li>- Applicare il concetto di variabile nella soluzione di schemi logici.</li> </ul> <p style="text-align: center;"><b>LEGGERE E SCRIVERE IL CODICE</b></p> <ul style="list-style-type: none"> <li>- Identificare istruzioni condizionali;</li> <li>- Creare istruzioni condizionali;</li> <li>- Utilizzare espressioni condizionali, cicli, funzione, variabile, ripetizione;</li> <li>- Debugging;</li> </ul>	<ul style="list-style-type: none"> <li>- Imparare cos’è un ciclo, quando usarlo;</li> <li>- Conoscere le espressioni condizionali se/allora;</li> <li>- Conoscere il concetto di funzione;</li> <li>- Conoscere il concetto di variabile;</li> <li>- Conoscere la funzione di ripetizione;</li> <li>- Conoscere le parti di un device;</li> <li>- Conoscere le principali funzioni di un programma ed utilizzarle in modo congruo;</li> <li>- Conoscere l’ambiente di programmazione visuale Scratch.</li> </ul>

	<ul style="list-style-type: none"> <li>- Scrivere codice conciso;</li> <li>- Apportare modifiche al codice;</li> <li>- Identificare i bug nel codice e dedurre soluzioni corrette.</li> </ul>	
<b>Contenuti</b>		
<ul style="list-style-type: none"> <li>- Concetto di programmazione (programmatore, linguaggio di programmazione, sequenza)</li> <li>- L'ora del codice</li> <li>- Code.org Corso 1 e 2, per le prime tre classi della primaria</li> <li>- Code.org Corso 3 e 4 per le ultime classi della primaria</li> <li>- Scratch (avvio al suo utilizzo)</li> <li>- Codi-Amo</li> <li>- Sequenza e Algoritmi, espressioni condizionali, cicli</li> <li>- Il labirinto</li> <li>- la funzione</li> <li>- la variabile</li> <li>- la ripetizione</li> <li>- Debugging</li> </ul>		

## Scuola Secondaria di Primo Grado

### Raccordi con la Scuola Primaria

- Conoscenza dei blocchi “Vai avanti”, “Gira a destra”; “Gira a sinistra”;
- L'Orsa del Codice di Programma il Futuro;
- Corso Rapido di 20 ore di Code.org
- Ambienti editor Scratch:

<b>Competenze</b>	<b>Abilità</b>	<b>Conoscenze</b>
<p>Sapersi esprimere e comunicare utilizzando codici e linguaggi diversi;</p> <p>L'alunno utilizza con consapevolezza i concetti di funzione, di variabile e di ripetizione</p> <p>Analizzare e rappresentare processi utilizzando modelli logici</p> <p>Approcciarsi ad un uso consapevole delle TIC</p> <p>Utilizzare trasversalmente le conoscenze</p>	<p style="text-align: center;"><b>COSTRUIRE ABILITA' DI BASE</b></p> <ul style="list-style-type: none"> <li>- Indicare la destra e la sinistra rispetto al personaggio da muovere sullo schermo;</li> <li>- Imparare a scrivere i comandi in ordine e risolvere i problemi utilizzando algoritmi;</li> <li>- Utilizzare il blocco di ripetizione;</li> <li>- Riconoscere in una dichiarazione logica sia l'ipotesi che la conclusione;</li> <li>- Utilizzare le espressioni condizionali se/allora;</li> <li>- Applicare il concetto di funzione nella soluzione di schemi logici;</li> <li>- Applicare il concetto di variabile nella soluzione di schemi logici.</li> </ul> <p style="text-align: center;"><b>LEGGERE E SCRIVERE IL</b></p>	<ul style="list-style-type: none"> <li>- Imparare cos'è un ciclo, quando usarlo;</li> <li>- Conoscere le espressioni condizionali se/allora;</li> <li>- Conoscere il concetto di funzione;</li> <li>- Conoscere il concetto di variabile;</li> <li>- Conoscere la funzione di ripetizione;</li> <li>- Conoscere le parti di un device;</li> <li>- Conoscere le principali funzioni di un programma ed utilizzarle in modo congruo;</li> <li>- Conoscere l'ambiente di programmazione visuale Scratch.</li> </ul>

	<p style="text-align: center;"><b>CODICE</b></p> <ul style="list-style-type: none"> <li>- Identificare istruzioni condizionali;</li> <li>- Creare istruzioni condizionali;</li> <li>- Utilizzare espressioni condizionali, cicli, funzione, variabile, ripetizione;</li> <li>- Debugging;</li> <li>- Scrivere codice conciso;</li> <li>- Apportare modifiche al codice;</li> </ul> <p style="text-align: center;">Identificare i bug nel codice e dedurre soluzioni corrette.</p>	
<b>Contenuti</b>		
<ul style="list-style-type: none"> <li>● Il linguaggio delle cose: inventiamo oggetti “smart”</li> <li>● Corso rapido di 20 ore di Code.org</li> <li>● Creazione di un video gioco con Code.org</li> <li>● Remix di progetti nati con Scratch</li> <li>● Ambienti editor Scratch</li> </ul>		
<b>TEMPI</b>	<p>L’organizzazione copre l’intero anno scolastico. Da ottobre a fine gennaio i contenuti, le metodologie e le attività proposte saranno strettamente collegati al pensiero computazionale. Nel secondo quadrimestre si lavorerà essenzialmente in previsione delle prove standardizzate nazionali di maggio, con esercizi volti ad abituare gli alunni al particolare format della Prova Invalsi.</p>	
<b>SOLUZIONI ORGANIZZATIVE</b>	<ul style="list-style-type: none"> <li>• Gruppi di classe in situazione frontale .</li> <li>• Gruppi omogenei</li> <li>• Gruppi di recupero/ d’approfondimento.</li> <li>• Lavori individuali</li> <li>• Momenti collegiali d’incontro con esperti esterni, in modalità online.</li> </ul>	
<b>ATTIVITA’</b>	<ul style="list-style-type: none"> <li>• <b>Unplugged:</b> tradizionali ( con uso di carta e penna): costruzione di labirinti, percorsi, disegni “pixati”...</li> <li>• <b>Tecnologiche:</b> sfruttando il collegamento a Internet e utilizzando piattaforme accreditate come “ Programma il futuro”.</li> </ul>	
<b>STRUMENTI</b>	<p><b>Unplugged:</b> CodyRoby, CodyWay.</p> <p><b>Tecnologici:</b>Code.org, Programma il futuro, Scratch.</p>	
<b>CONTROLLO DEGLI APPRENDIMENTI</b>	<p>Verifica e Valutazione delle competenze mediante:</p> <ol style="list-style-type: none"> <li>1. Gli stessi elaborati prodotti</li> <li>2. Osservazioni sistematiche</li> <li>3. Situazioni colloquiali di vario tipo</li> <li>4. Prove a scelta multipla (format Invalsi) per la seconda parte</li> </ol>	



	dell'anno.
<b>PARTECIPAZIONE AD EVENTI COLLETTIVI</b>	<ol style="list-style-type: none"><li>1. 15/23 Ottobre 2016 partecipazione alla "Europe Code Week"</li><li>2. 5/11 Dicembre partecipazione alla "Hour of Code"</li></ol>