

# **CURRICOLO VERTICALE DI CODING E ROBOTICA EDUCATIVA a.s. 2019/2020**

## **INTRODUZIONE**

Entro il 2022 il governo si è impegnato ad introdurre, progressivamente e gradualmente, lo sviluppo del pensiero computazionale e lo studio del coding nel curricolo digitale obbligatorio della scuola dell'infanzia e primaria e a promuovere e favorire iniziative volte all'alfabetizzazione e allo sviluppo dell'apprendimento del «coding» nelle scuole secondarie di primo e secondo grado. (Mozione n.1-00117 del 12 marzo 2019). In realtà il Miur aveva già riconosciuto la validità didattica del Coding con la Nota 2397 del 23 settembre 2014, con la quale aveva avviato l'iniziativa "Programma il futuro", con l'obiettivo di fornire alle scuole una serie di strumenti semplici, divertenti e facilmente accessibili per formare gli studenti ai concetti base dell'informatica attraverso la programmazione in un contesto di gioco (coding). Tale iniziativa è stata rinnovata anche nell'anno scolastico successivo con la Circolare 9759 dell'8 ottobre 2015. Come previsto anche nel Piano Nazionale Scuola Digitale, (L. 107/2015) un'appropriata educazione al "pensiero computazionale", che vada al di là dell'iniziale alfabetizzazione digitale, è infatti essenziale affinché le nuove generazioni siano in grado di affrontare la società del futuro non da consumatori passivi ed ignari di tecnologie e servizi, ma da soggetti consapevoli di tutti gli aspetti in gioco e come attori attivamente partecipi del loro sviluppo. Introdurre l'insegnamento del Coding, fin dalla scuola dell'infanzia e primaria, è fondamentale per favorire nei discenti la formazione su tematiche centrali, quali il pensiero computazionale, la creatività digitale e la cittadinanza digitale, in quanto al di là delle competenze specifiche (che il coding e la programmazione portano con sé) il vero valore aggiunto consiste proprio nel metodo che queste nuove discipline consentono di acquisire, metodo finalizzato a stimolare e rafforzare la capacità di pensiero logico e creativo, di analisi e di risoluzione dei problemi.

## **CODING: ATTIVITA' TRASVERSALE**

Il coding viene spesso associato solo ed esclusivamente alla tecnologia e più strettamente all'informatica. Ma relegare il coding al solo ambito informatico è riduttivo e semplicistico. Andrebbe invece visto come uno strumento di tipo trasversale che accompagna l'allievo nel maturare il proprio pensiero logico nel percorso di crescita quotidiano. Il coding dunque non è solo un'attività laboratoriale dell'ora di tecnologia, ma va considerato come approccio all'analisi, alla logica e alla codifica di tutte quelle informazioni che fanno parte di un processo formativo.

## **DAL CODING ALLA ROBOTICA EDUCATIVA**

Un altro strumento molto efficace allo sviluppo del pensiero computazionale è la robotica educativa. Essa è la concretizzazione di quanto "progettato" con il coding. I robot devono seguire le istruzioni impartite attraverso un programma (codice o code). Si parla di robotica educativa e

non di robotica, perché non si tratta di aggiungere un'altra materia di studio alle tante che già ci sono. Si tratta invece di utilizzare la costruzione e la programmazione di robot all'interno di un percorso di lavoro ben definito, per aiutare gli alunni a raggiungere le competenze previste.

La robotica educativa è un approccio nuovo all'insegnamento, un metodo che utilizza i robot per stimolare la curiosità e l'uso della logica e del ragionamento nei bambini e nei ragazzi, così imparano a risolvere piccoli problemi di difficoltà crescente mentre si divertono. La robotica educativa promuove le attitudini creative degli studenti, nonché la loro capacità di comunicazione, cooperazione e lavoro di gruppo. Essa può diventare il modo più semplice per creare un ambiente di apprendimento innovativo, creativo e divertente poiché è in grado di coinvolgere attivamente gli studenti nelle lezioni, aumentando il loro interesse per l'ambiente scolastico. Favorisce il dialogo, la comunicazione, il confronto attivo degli studenti su tematiche curriculari e non, agevolandone l'integrazione e la capacità di relazione e comunicazione. L'elaborazione di un processo complesso obbliga gli studenti a sviluppare il proprio pensiero critico e ad imparare ad esporre il proprio lavoro a compagni ed insegnanti. La robotica è una disciplina ampiamente riconosciuta e apprezzata nel contesto della dispersione scolastica e dell'inclusione giovanile. Inoltre, il naturale appeal che i robot esercitano sui ragazzi, rende il processo di apprendimento più divertente ed appagante, permettendo di costruire un percorso stimolante, perfetto per motivare anche gli studenti meno inseriti nel contesto scolastico.

### **COMPETENZE CHIAVE EUROPEE (aggiornate alla Nuova Raccomandazione del Consiglio UE del 22 maggio 2018)**

- competenza alfabetica funzionale;
- competenza multilinguistica;
- competenza matematica e competenza in scienze, tecnologie e ingegneria;
- competenza digitale;
- competenza personale, sociale e capacità di imparare ad imparare;
- competenza in materia di cittadinanza;
- competenza imprenditoriale;
- competenza in materia di consapevolezza ed espressione culturali.

### **CURRICOLO SCUOLA DELL'INFANZIA**

Coding e pensiero computazionale rendono possibile apprendere strategie di risoluzione di problemi, progettazione e comunicazione anche a chi non programmerà mai.

Attraverso il coding si cerca di trasmettere ai “nativi digitali” sia il linguaggio della programmazione e degli algoritmi (mettendo in primo piano la logica che sta dietro la tecnologia) sia la potenza della segmentazione di un problema e il loro affidamento a piccoli gruppi, così da dare forza al lavoro collaborativo, al rispetto delle diverse idee.

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 problema.

Il coding unplugged è particolarmente adatto alla didattica della Scuola dell’Infanzia, perché stimola l’apprendimento e la curiosità nei bambini; ogni situazione-problema che preveda per la sua risoluzione una procedura da seguire può definirsi coding unplugged, ovvero un’attività che utilizza strumenti non digitali per la sua realizzazione. Poiché nelle attività scolastiche il pensiero computazionale non si ricollega sempre alla presenza del computer, lo si sviluppa attraverso attività reali e pratiche: normalmente vengono realizzati grandi reticoli da mettere a terra in cui i bambini possono fare veri e propri giochi di ruolo, diventando protagonisti del loro apprendimento.

In questo tipo di attività, ci sono due personaggi: il programmatore e il robot che esegue i comandi; entrambi possono essere bambini e il programmatore guida il robot, utilizzando un linguaggio simbolico di programmazione (ad esempio le quattro frecce direzionali).

Il percorso di solito viene disegnato su un foglio e riproduce il reticolo rappresentato a terra. Il robot ascolta i comandi di movimento e li esegue camminando sul reticolo. La scrittura del programma può avvenire in due modi: all’interno della griglia che l’alunno ha in mano, utile per i bambini più piccoli, oppure accanto alla griglia stessa.

<b>COMPETENZE</b>	<b>OBIETTIVI DI APPRENDIMENTO</b>
<ul style="list-style-type: none"> <li>- Saper trovare e utilizzare strategie condivise;</li> <li>- Saper collaborare e condividere con il gruppo per il raggiungimento di un obiettivo comune;</li> <li>- Saper creare “strisce di comandi”;</li> <li>- Saper prevedere situazioni e conseguenze.</li> </ul>	<ul style="list-style-type: none"> <li>- Consolidare il concetto di destra e sinistra;</li> <li>- Sviluppare la percezione spaziale;</li> <li>- Padroneggiare la complessità;</li> <li>- Mettere in atto strategie risolutive;</li> <li>- Ipotizzare un percorso;</li> <li>- Dare istruzioni;</li> <li>- Osservare, descrivere e progettare percorsi secondo vincoli dati;</li> <li>- Fare esperienza di lavoro di gruppo.</li> </ul>

<b>CLASSE</b>	<b>METODOLOGIA</b>	<b>STRUMENTI E CONTENUTI</b>	<b>VERIFICHE</b>
<b>SEZIONI</b>	Favorire la curiosità, la scoperta, l’esplorazione concreta, il gioco, il	<ul style="list-style-type: none"> <li>- Giochi su scacchiere di piccole e grandi dimensioni</li> </ul>	Prodotti realizzati

<b>ALUNNI 5 ANNI</b>	procedere per tentativi, la collaborazione, la riflessione sulle esperienze.	<ul style="list-style-type: none"><li>- Pixel Art</li><li>- CodyFeet</li></ul> Robotica: <ul style="list-style-type: none"><li>- DOC</li><li>- CodaBruco</li></ul>	
<b>Eventi a cui è possibile partecipare: Europe CodeWeek (ottobre)</b>			

## CURRICOLO SCUOLA PRIMARIA

L'approccio metodologico prevede l'utilizzo di strumenti a difficoltà progressiva, sia per le attività tecnologiche, sia per quelle unplugged e di robotica.

COMPETENZE DISCIPLINARI	OBIETTIVI DI APPRENDIMENTO
<ul style="list-style-type: none"> <li>- Sapersi esprimere e comunicare utilizzando codici e linguaggi diversi</li> <li>- Analizzare e rappresentare processi utilizzando modelli logici</li> <li>- Utilizzare trasversalmente le conoscenze</li> <li>- Definire soluzioni flessibili per problemi complessi</li> <li>- Applicare le conoscenze esistenti per generare nuove idee e prodotti</li> <li>- Maturare la consapevolezza delle potenzialità, dei limiti e dei rischi dell'uso delle TIC</li> <li>- Generalizzare una soluzione e adattarla ad altri ambiti</li> </ul>	<ul style="list-style-type: none"> <li>- Apprendere in modo intuitivo e ludico i concetti base della programmazione per sviluppare il pensiero computazionale</li> <li>- Acquisire il modello del pensiero computazionale come metodo per la risoluzione dei problemi</li> <li>- Saper applicare le modalità operative del coding trasversalmente e nelle situazioni problematiche della vita quotidiana</li> <li>- Favorire l'utilizzo diretto di conoscenze matematiche, linguistiche, antropologiche e scientifiche</li> <li>- Collaborare con altri per la soluzione di problemi</li> <li>- Riflettere sul lavoro svolto valutandolo secondo determinati criteri</li> <li>- Lavorare in team e comunicare con gli altri</li> <li>- Modellare la realtà ed individuare algoritmi</li> <li>- Approcciarsi ad un uso consapevole delle TIC</li> <li>- Sviluppare il pensiero creativo, riflessivo e procedurale</li> <li>- Indurre alla riflessione sull'errore</li> </ul>

CLASSE	METODOLOGIA	STRUMENTI E CONTENUTI (Da selezionare tra quelli di seguito elencati)	VERIFICHE
1 <sup>^</sup> -2 <sup>^</sup> -3 <sup>^</sup> PRIMARIA	Approccio mediato dal docente Learning by doing	Introduzione ai concetti di programmazione (linguaggio delle cose, istruzioni, sequenze, algoritmi...) <b>Unplugged:</b> CodyRoby	Prodotti realizzati

	<p>Cooperative Learning          Problem solving          Didattica dell'errore          (debugging)          Tutoring</p>	<p>CodyColor          CodyWay          CodyWords          Pixel art          Attività di Programma il futuro:          - Pensiero computazionale          - Programmazione su carta quadrettata          - Algoritmi          - Cittadinanza digitale consapevole  <b>Tecnologiche:</b>          Ora del codice          Attività su Code.org:          - Corso 1          - Corso 2          Scratch Jr.          Scratch 3.0 (blocchi: avanti, gira a destra, gira a sinistra, ripetizione, condizione (se... allora, altrimenti)          Altri programmi online: Codemonkey, Lightbot, Run Mario, Kodable, Tynker...  <b>Robotica educativa:</b>          kit Lego we.do 2.0          Ozobot Evo          Altri Robot didattici</p>	
<p><b>4<sup>^</sup>-5<sup>^</sup>          primaria</b></p>		<p>Introduzione ai concetti di programmazione (linguaggio delle cose, istruzioni, sequenze, algoritmi...)  <b>Unplugged:</b>          CodyRoby          CodyColor          CodyWay          CodyWords          Pixel art</p>	

		<p>Pixel art: Codice esadecimale</p> <p>Attività di Programma il futuro:</p> <ul style="list-style-type: none"> <li>- Funzioni</li> <li>- Istruzioni condizionali</li> <li>- Composizione di canzoni</li> <li>- Astrazione</li> <li>- Programmazione a staffetta (debugging)</li> <li>- Cittadinanza digitale consapevole</li> <li>- Come funzionano i computer</li> </ul> <p><b>Tecnologiche:</b></p> <p>Ora del codice</p> <p>Attività su Code.org:</p> <ul style="list-style-type: none"> <li>- Corso 3</li> <li>- Corso 4</li> </ul> <p>Scratch 3.0 (blocchi: avanti, gira a destra, gira a sinistra, ripetizione, condizione (se... allora, altrimenti), sensori, variabili e liste, funzioni e parametri)</p> <p>Altri programmi online: Codemonkey, Lightbot, Run Mario, Kodable, Tynker...</p> <p><b>Robotica educativa:</b></p> <p>kit Lego we.do 2.0</p> <p>Ozobot Evo</p> <p>Altri Robot didattici</p>	
<p><b>Eventi a cui è possibile partecipare:</b></p> <p><b>European CodeWeek (ottobre)</b></p> <p><b>Hour of Code (Dicembre)</b></p> <p><b>Safer Internet Day (febbraio)</b></p> <p><b>Festa del PNSD (aprile)</b></p>			

# CURRICOLO SCUOLA SECONDARIA DI PRIMO GRADO

## Prerequisiti:

- Conoscenza dei blocchi “Vai avanti”, “Gira a destra”; “Gira a sinistra”; “Ripeti”; “Se – Allora – Altrimenti”
- L’Ora del Codice di Programma il Futuro;
- Code.org Corso 2 e 3
- Ambiente editor Scratch Jr
- Ambiente sviluppo Scratch: sprite, stage, costume e script modifica sprite con editor grafico inizializzazione di un progetto

Nella Scuola Secondaria di Primo Grado fondamentale risulta essere l’interdisciplinarietà del pensiero computazionale. Non si programmerà l’ora di coding, ma si utilizzeranno le modalità che lo contraddistinguono per descrivere attività e procedure e per concettualizzare procedimenti e soluzioni. Il pensiero computazionale sarà svincolato dagli ambiti prettamente tecnici per rientrare nell’attività didattica di più discipline, sarà uno strumento che facilita la comprensione dei contenuti.

Il coding sarà condiviso con gli studenti e non proposto dai docenti, sarà sperimentato attivamente con loro e non progettato e pensato per loro. L’insegnante non offrirà alla classe una soluzione, ma un problema giusto e reale a cui applicare il coding. Si impara quindi facendo, in un’ottica interdisciplinare, dove ci si può permettere di mescolare creatività e fantasia con la logica.

Validi raccordi saranno possibili in:

- *Matematica e Geometria, Tecnologia* (disegno tecnico) - Problem solving, logica, capacità di trasformare le proprie idee in App e Videogiochi utilizzando i più conosciuti linguaggi di programmazione per bambini e ragazzi, come Scratch.
- *Italiano e Lingue straniere* - Realizzazione di Storytelling con ambienti di sviluppo quali Scratch, realizzazione di app in lingua straniera.

COMPETENZE DISCIPLINARI	OBIETTIVI DI APPRENDIMENTO
<ul style="list-style-type: none"><li>- Conoscere il concetto di algoritmo nelle sue varie forme ed applicazioni</li><li>- Approfondire i concetti di variabile e di funzione con parametri</li><li>- Conoscere le caratteristiche di un robot</li><li>- Utilizzare trasversalmente le conoscenze</li></ul>	<p>COSTRUIRE ABILITA’ DI BASE</p> <ul style="list-style-type: none"><li>- Imparare a scrivere algoritmi in maniera corretta e a risolvere problemi complessi scomponendoli in problemi più semplici.</li><li>- Applicare il concetto di variabile nella soluzione di schemi logici</li><li>- Utilizzare correttamente le funzioni con e senza parametri</li></ul> <p>LEGGERE E SCRIVERE IL CODICE</p> <ul style="list-style-type: none"><li>- Debugging</li></ul>



	<ul style="list-style-type: none"> <li>- Progettare e costruire storie tecnologiche unendo con il linguaggio di programmazione Scratch.</li> </ul> <p><b>COSTRUIRE E PROGRAMMARE UN ROBOT</b></p> <ul style="list-style-type: none"> <li>- Capire cosa sono e come usare sensori e motori per rendere interattivi i modelli</li> <li>- Stabilire relazioni causa-effetto</li> <li>- Creare semplici programmi per istruire i modelli LEGO</li> </ul> <p><b>PROBLEM SOLVING</b></p> <ul style="list-style-type: none"> <li>- Saper lavorare in gruppo.</li> <li>- Ottimizzare la soluzione di un problema</li> <li>- Scomporre un problema complesso in problemi di complessità inferiore</li> <li>- Utilizzare registri linguistici adatti agli interlocutori</li> <li>- Utilizzare le conoscenze disciplinari per riempire di contenuti le attività: realizzare storytelling; conoscere le principali regole nella realizzazione di figure geometriche ecc. ecc.</li> </ul>
--	--

<b>CLASSE</b>	<b>METODOLOGIA</b>	<b>STRUMENTI E CONTENUTI</b> <b>(Da selezionare tra quelli di seguito elencati)</b>	<b>VERIFICHE</b>
<b>1<sup>^</sup>-2<sup>^</sup>-3<sup>^</sup> SECONDARIA</b>	Approccio mediato dal docente Learning by doing Cooperative Learning Problem solving Didattica dell'errore (debugging) Tutoring	<ul style="list-style-type: none"> <li>- L'ora del codice</li> <li>- Il linguaggio delle cose: inventiamo oggetti "smart"</li> <li>- Code.org</li> <li>- Approfondimento del concetto di Debug</li> <li>- Remix di semplici progetti nati con Scratch</li> <li>- Ambienti di sviluppo Scratch: Pixel e coordinate dello schermo <ul style="list-style-type: none"> <li>○ Aspetto e utilizzo dei costumi</li> <li>○ Controllo</li> </ul> </li> </ul>	Prodotti realizzati

		<ul style="list-style-type: none"> <li>○ Movimento</li> <li>○ Sensori (attendi, sta toccando)</li> <li>○ Stage ed utilizzo degli sfondi</li> <li>○ Utilizzo di Variabili e Funzioni</li> <li>○ Messaggi (comunicazione tra gli sprite)</li> <li>○ Creazione di uno Storytelling con Scratch</li> </ul> <p><b>Oggetti e loro metodi</b></p> <p><b>Sensori e Timer</b></p> <ul style="list-style-type: none"> <li>- Utilizzo software di progettazione Lego Mindstorm (costruire un robot)</li> <li>- Utilizzo software di programmazione Lego Mindstorm (programmare un robot):</li> </ul> <p><b>Motori e sensori</b></p> <p><b>Costrutti principali utilizzati nella programmazione dei robot</b></p> <ul style="list-style-type: none"> <li>- Realizzazione di un proprio robot Lego Mindstorm da far interagire con altri robot simili</li> </ul>	
<p><b>Eventi a cui è possibile partecipare:</b>  <b>European CodeWeek (ottobre)</b>  <b>Hour of Code (Dicembre)</b>  <b>Safer Internet Day (febbraio)</b>  <b>Festa del PNSD (aprile)</b></p>			

## SITOGRAFIA

Programma il futuro: Lezioni unplugged - <https://programmailfuturo.it/come/lezioni-tradizionali/>

Programma il futuro: Lezioni tecnologiche - <https://programmailfuturo.it/come/lezioni-tecnologiche/>

Ora del codice - <https://programmailfuturo.it/come/ora-del-codice> - <https://hourofcode.com/us/it/learn>

Code.org: Corso 1 - <https://studio.code.org/s/course1>

Code.org: Corso 2 - <https://studio.code.org/s/course2>

Code.org: Corso 3 - <https://studio.code.org/s/course3>

Code.org: Corso 4 - <https://studio.code.org/s/course4>

Scratch Junior - <https://www.scratchjr.org/>

Scratch 3.0 - <https://scratch.mit.edu/>

CodyRoby - <http://codemooc.org/codyroby/>

CodyWay - <http://codeweek.it/lm05-codyway/>

CodyColor - <http://codemooc.org/codycolor/>

CodyColor online - <https://codycolor.codemooc.net/#/>

Pixel art - <http://codemooc.org/pixel-art/>

Codice esadecimale - <http://codemooc.org/pixel-art-esadecimale/>

CodyWords - <http://codemooc.org/codicando-o-codywords/>

CodyFeet - <http://codemooc.org/codyfeet/>