

CODING

**STRUMENTO DI SVILUPPO
COGNITIVO**

*Annamaria Brambilla
J.C. Salvo D'Acquisto - Muggiò*



Dal pensiero computazionale al coding

Seymour Papert

(Pretoria, 29 febbraio 1928 – Blue Hill, 31 luglio 2016) matematico, informatico e pedagogo sudafricano naturalizzato statunitense.

- 📖 Lavorò con Jean Piaget
- 📖 Si trasferì negli anni sessanta al MIT per occuparsi di intelligenza artificiale.
- 📖 Fu uno dei fondatori del costruzionismo.
Per approfondire: [Costruzionismo](#), La Scuola



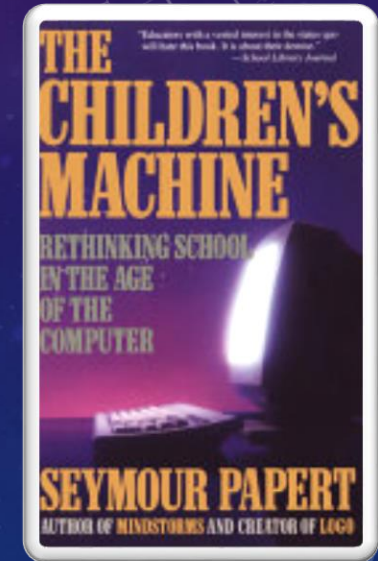
Dal pensiero computazionale al coding

“The role of the teacher is to create the conditions for invention rather than provide ready-made knowledge”

The children's machines: Rethinking schools in the age of the computer, 1993

Il ruolo dell'insegnante è creare le condizioni perché avvengano le scoperte piuttosto che fornire conoscenze preconfezionate

- L'apprendimento avviene in modo più efficiente se chi apprende è coinvolto nella produzione e nell'interazione con oggetti;
- il bambino apprende con l'aiuto di **artefatti cognitivi**, oggetti o dispositivi che facilitano l'apprendimento
- negli ambienti per l'apprendimento che utilizzano tecnologie, l'allievo ha un ruolo centrale perché sono le sue attività che generano apprendimenti

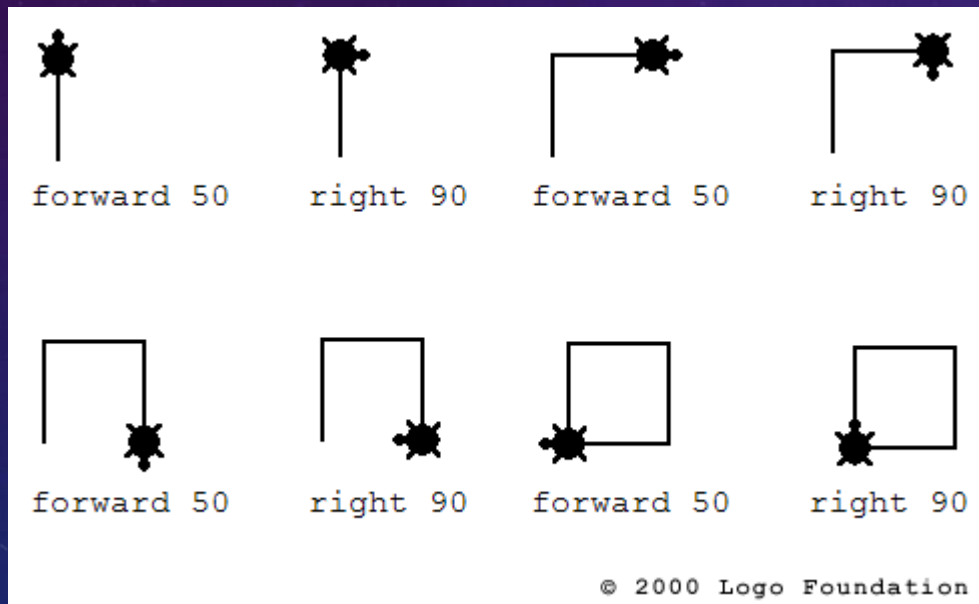


[La Scuola: La nuova didattica](#)
[Indire: Apprendimento cooperativo](#)

Realizza il **linguaggio LOGO** per consentire ai bambini *di controllare* il computer.

In quest'ambiente, il docente si trasforma in animatore della comunità, promotore di attività in cui i bambini progettano e imparano, verbalizzando ciò che apprendono e discutendo.

...l'obiettivo è insegnare in modo tale da offrire il maggiore apprendimento con il minimo di insegnamento.



LEGOMindstorm: <http://mindstorms.lego.com/en->

Dal pensiero computazionale al coding

Jeannette M. Wing

Vicepresidente di Microsoft Research; è stata Professor of Computer Science at Carnegie Mellon University, Pittsburgh, Pennsylvania, United States; scienziata informatica Jeannette Wing nel 2006 rese popolare l'espressione

["Pensiero computazionale"](#) (in inglese)



[By Ronald Woan](#)

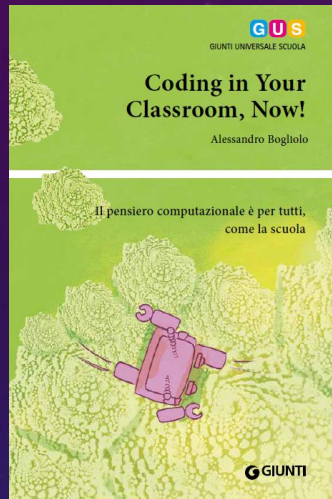
Attribution-NonCommercial 2.0 Generic (CC BY-NC 2.0)

Dal pensiero computazionale al coding

Thinking like a computer scientist means more than being able to program a computer. It requires thinking at multiple levels of abstraction.

- Conceptualizing, not programming.
- Fundamental, not rote skills
- A way that humans, not computers, think
- Complements and combines mathematical and engineering thinking
- Ideas, not artifacts
- For everyone, everywhere
- Intellectually challenging and engaging
scientific problems remain to be understood and solved
- One can major in computer science and do anything

Dal pensiero computazionale al coding



Il termine Coding si riferisce all'uso di strumenti e metodi di **programmazione visuale a blocchi** per favorire lo sviluppo del pensiero computazionale.

«Ma se avessi scelto di tradurre coding in italiano che termini avrei trovato? Codifica, scrittura di codice, cifratura, programmazione, ... Come vedi non avevo scelta, perché ciò di cui ti voglio parlare non è questo. Non è roba da programmatori né da agenti segreti.»

Alessandro Bogliolo, docente di Architettura degli elaboratori all'Università di Urbino

Il pensiero computazionale è **un processo mentale** per la risoluzione di problemi costituito dalla combinazione di metodi caratteristici e di strumenti intellettuali, entrambi di valore generale.

(www.programmailfuturo.it)

Fare **coding** quindi

«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» Alessandro Bogliolo

«I benefici del “*pensiero computazionale*” si estendono a tutte le professioni.

Avvocati, insegnanti, dirigenti di azienda, psicologi, architetti, medici, funzionari di amministrazioni - solo per citare alcune professioni - ogni giorno devono

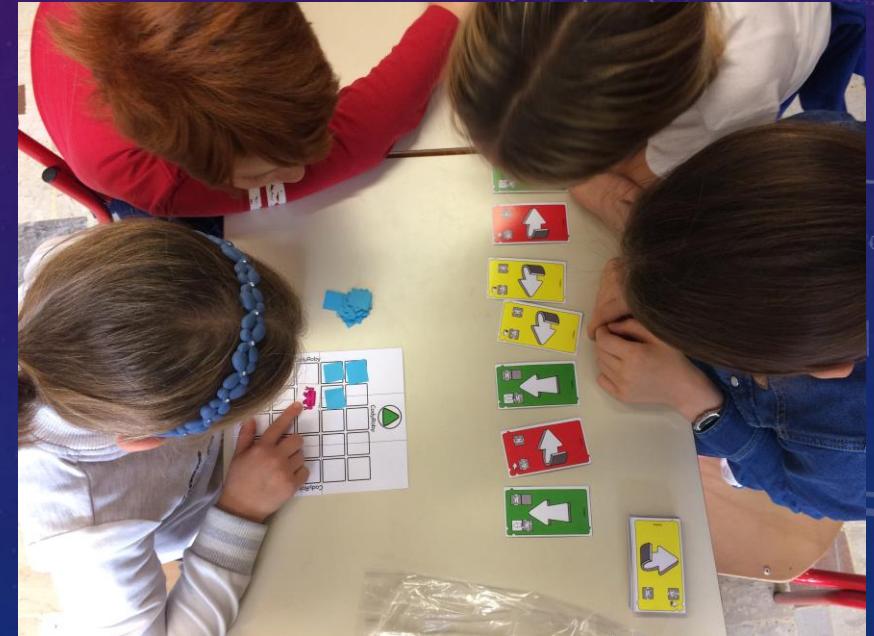
- affrontare problemi complessi;
- ipotizzare soluzioni che prevedono più fasi e la collaborazione con altri colleghi o collaboratori;
- formulare una descrizione chiara di cosa fare e quando farlo.» [link a Programma il futuro](#)

Coding in classe

E' possibile sviluppare il pensiero computazionale attraverso:

- lezioni tradizionali, *unplugged*, che non richiedono l'utilizzo di dispositivi digitali
- Lezioni tecnologiche.

In rete è possibile trovare numerose risorse.



Attività Unplugged

Dal sito [Codeweek](#) è possibile scaricare il [kit per realizzare il gioco Cody Roby](#) - ([CodyRoby - en](#)) - e [Cody Way](#) per i ragazzi più grandi.



Cody&Roby #1. [Seguimi](#)



Coding in classe

Programma il futuro è un progetto del MIUR realizzato con la collaborazione del CINI (Consorzio Interuniversitario Nazionale per l'Informatica).



Iscrivendosi come insegnanti è possibile procedere a:

- Creazione della classe
- Inserimento di studenti nella classe
- Accesso degli studenti alle lezioni
- Inizio delle attività
- Attestati di svolgimento

Il sito mette a disposizione materiali in italiano per realizzare alcune attività unplugged, lezioni tradizionali, che non richiedono l'utilizzo di dispositivi elettronici.

Coding in classe

Le attività del progetto **Programma il futuro** si articolano su due siti web:

1. il [sito di supporto](#), con una funzione di guida e di approfondimento;
2. il [sito di fruizione](#) delle lezioni: Code.org.

Sul sito di code.org è possibile scaricare **il [curriculum](#)** per sviluppare alcuni importanti concetti: algoritmo, sistema binario



Pensiero computazionale

These tips will help you get unstuck when solving Code.org puzzles!

Step 1: Understand the Puzzle

- What does the puzzle want you to do?
- Can you talk about the problem in your own words?
- Were you given any code to start?
 - What does it do?
 - Why do you think it's there?
- What is the goal of the puzzle?
- Have you solved any other puzzles that are like this one?

Step 2: Create a Plan (Pick one or more)

- Write an algorithm.
- Guess and check as you go.
- Draw a picture of what you want to do.
- Try working backward.
- Solve one small piece at a time.
- Compare to a puzzle that you've already solved.

Step 3: Perform and Perfect the Plan

- Did you solve the puzzle?
- If not, hunt for one error at a time.
- Retest your plan after every change.
- If you start to get frustrated, take a deep breath, or leave your screen for a minute. When you come back, you may see what was causing the trouble!
- Ask questions. Maybe one of your friends can help you figure out where your plan goes awry.

Step 4: Check Your Work

- Does your answer solve the puzzle?
- Did you hit all of the goals of this puzzle?
- Now that you have one way to solve the puzzle, is there an easier way to do it?
- If you change this solution a little, will it work for any other puzzles?
- Could you explain your solution to help

Pensiero computazionale

G

Code.org Puzzle Solving Recipe

(Based on Polya's Four Step Problem Solving Process)

C O
D E

Talk about these steps with your students so they have tools to get themselves unstuck while working on puzzles. Also, use these steps to ask leading questions to students who ask for help.

Step 1: Understand the Puzzle

- * Do you understand the situation or puzzle prompt?
- * Can you restate the problem in your own words?
- * Do you understand the code you are given and why?
 - What role does the code play?
- * Do you know what the goal of the puzzle is?
- * Is this problem similar to another puzzle you have solved?

Step 2: Create a Plan

- * Can one (or more) of the following strategies be used?
 - Guess and test
 - Draw a map
 - Draw a picture
 - Look for a pattern
 - Compare to a previously solved puzzle
 - Solve a simpler problem
- * Draw a diagram
- * Solve an equivalent problem
- * Identify subgoals
- * Work backwards

Pensiero computazionale

G

Code.org Puzzle Solving Recipe

(Continued)

C O
D E

Step 3: Perform and Perfect the Plan

* Attempt to solve the puzzle based on the plan that you devised. If the strategy did not succeed, look carefully at the feedback provided by any errors that were created, and modify your plan.

* Test and change your strategy often. Don't be afraid to try solutions before you know that they are perfect. You can often reach the right answer by using trial and error each step of the way.

* Take a walk. If you start to get frustrated, leave the screen. Look away. Think of something else for a little while. When you get back, the answer just might come to you!

* Talk with others. They may be able to give you hints, or even explain how something works so that you can discover a solution.

Step 4: Check Your Work

* Is your solution correct? Does your answer satisfy all goals (number of blocks, use of a particular block/concept)?

* Can you see an easier or more efficient solution?

* Can you see how you can extend your solution to a more general puzzle pattern?

Attività tecnologiche code.org

La piattaforma di Code.org mette a disposizione di docenti e studenti una serie di strumenti adatti ad ogni età; dagli esercizi per principianti ad App Lab, un ambiente di apprendimento per realizzare semplici applicazioni.

Dal sito è possibile scaricare un libro con le attività, 441 pagine in formato pdf, e un testo con le soluzioni dei diversi step.
(in inglese)



Lezione 4: Labirinto: Sequenze

STUDIO

Blocchi

Area di lavoro: 1 / 8 blocchi

quando si clicca su "Esegui"

Esegui

Percorsi tecnologici per tutte le età

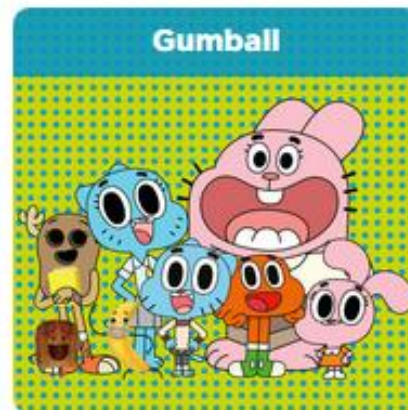
Attività tecnologiche

... in ogni disciplina

Story telling



Scegli il tema del tuo Laboratorio



Attività tecnologiche

Scratch è un progetto dei Media Lab del MIT.

La piattaforma è completamente gratuita e mette a disposizione una serie di strumenti per programmare storie interattive, giochi e animazioni e condividerle con gli altri membri della comunità.

«Scratch insegna ai giovani a pensare in maniera creativa, a ragionare in modo sistematico e a lavorare in maniera collaborativa — queste sono tutte capacità essenziali per chi vive nel 21mo secolo.»

A questo link è possibile trovare guide e tutorial:

- La pagina delle Cose da Provare offre una varietà di tutorial, schede attività e guide per gli educatori.
- La Finestra dei Suggerimenti fornisce aiuto per creare progetti con Scratch.
- La Guida al Curriculum per il Creative Computing fornisce piani, attività e strategie per introdurre i tuoi studenti al creative computing.

The Scratch logo is displayed in a stylized, bubbly font. The letters are orange with a white outline and a slight drop shadow, giving it a 3D appearance. The background of the slide features faint, circular technical diagrams and a starry space pattern.

Attività tecnologiche

Attività realizzate dai bambini e dai ragazzi dell'I.C. S. D'Acquisto di Muggiò

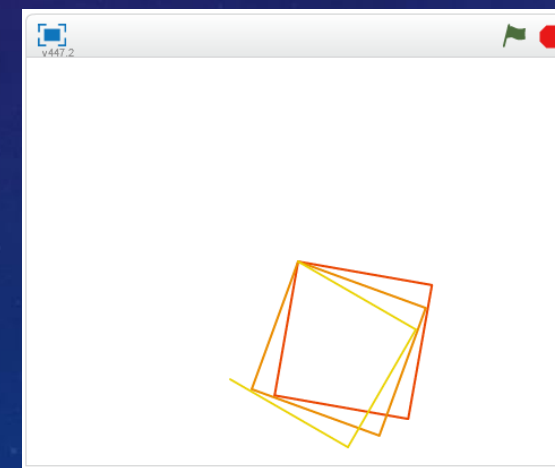
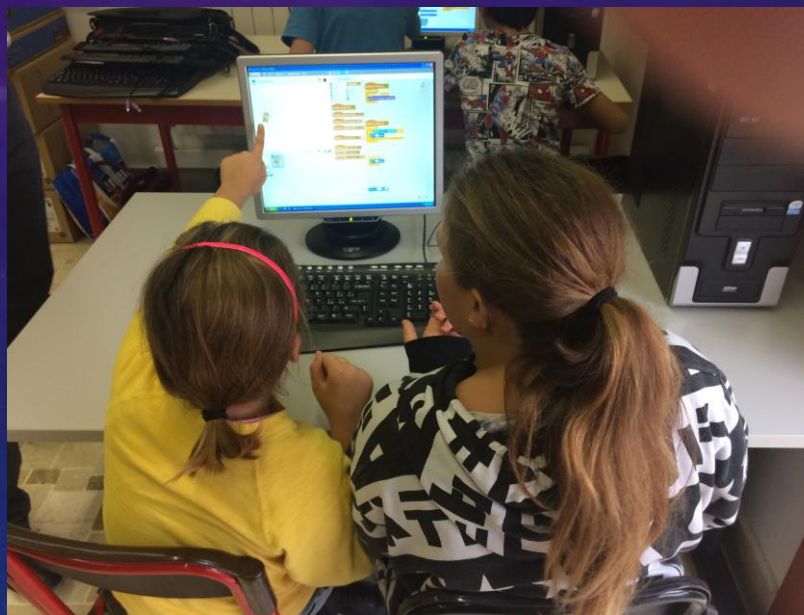
L'orto della signora Marta



Inventare giochi con Scratch



«Che bella la matematica!»



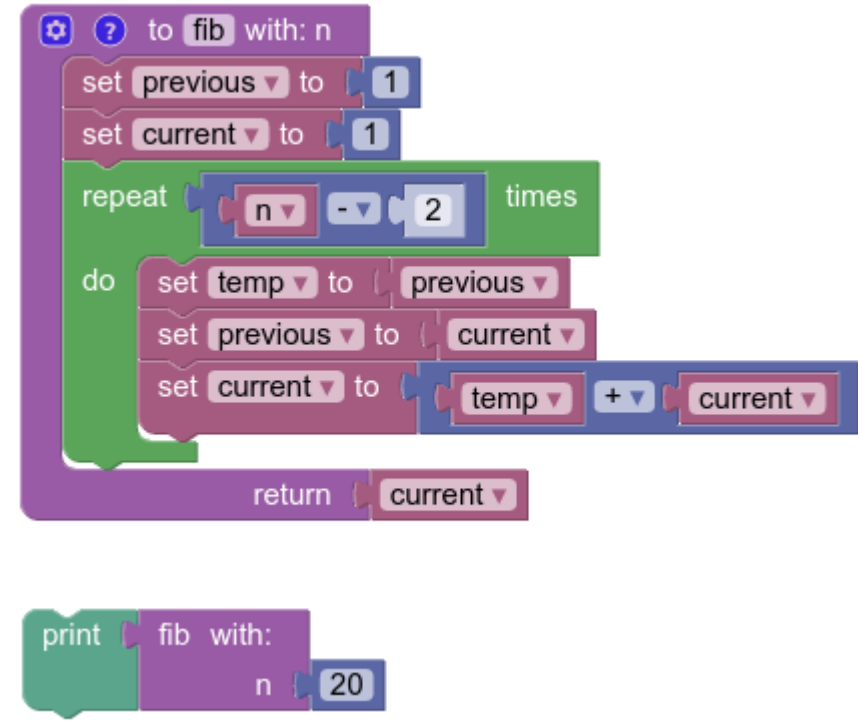
Google Blockly

Blockly è un ambiente di **programmazione visuale** che consente di apprendere le basi della programmazione in modo facile.

<https://developers.google.com/blockly/>

Robotiko: Blockly cos'è e come si usa

<https://www.robotiko.it/blockly-cose-come-funziona-come-si-usa/>



```
to fib with: n
  set previous to 1
  set current to 1
  repeat (n - 2) times
    do
      set temp to previous
      set previous to current
      set current to temp + current
  return current

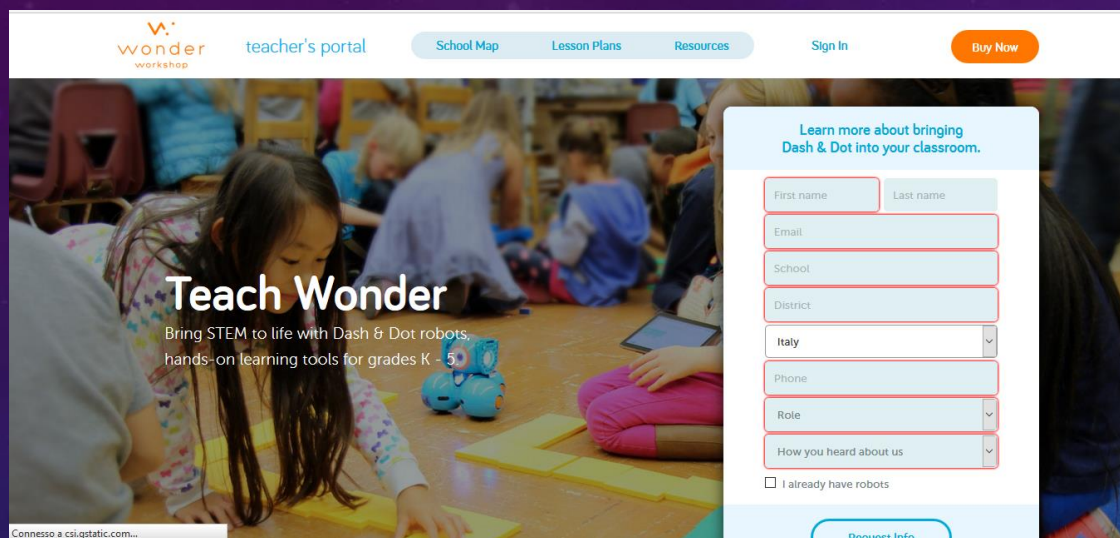
print fib with:
  n 20
```

The image shows a Blockly script for a Fibonacci function. The function is named 'fib' and takes an argument 'n'. It initializes 'previous' and 'current' to 1. A 'repeat' block loops 'n - 2' times. Inside the loop, a 'do' block contains three 'set' blocks: 'temp' is set to 'previous', 'previous' is set to 'current', and 'current' is set to 'temp + current'. After the loop, the function returns 'current'. Below the function definition, there is a 'print' block that calls 'fib with: n' where 'n' is set to 20.

Robotica, coding per parlare con le cose: i robot

[Wonder risorse per docenti](#)

[Arduino Education](#)



Dash and Dot possono essere controllati da 5 applicazioni gratuite, che si collegano ai robot via **bluetooth**, compatibili sia con i dispositivi iOS sia **Android**: Go, Path, Xylo, Wonder e Blockly

Amber Monza 2017



Arduino è una piattaforma hardware composta da una serie di schede elettroniche dotate di un microcontrollore.

Realtà virtuale

Che cos'è [WildCard](#)

[Articolo di Focus](#)

Ricerca: con occhiali 'magici' terapia virtuale per bimbi disabili
Dal politecnico Milano progetto per piccoli colpiti da autismo, ritardo psicomotorio, sindrome Down



Formazione

In rete si trovano occasioni per la propria formazione.

Coding in your classroom, Now! è forse quella che continua a riscuotere successo.

Le iscrizioni sono sempre aperte.

<http://codemooc.org/>

EMMA

1506
UNIVERSITA
DEGLI STUDI
DI URBINO
CARLO BO

#CODE
MOOC

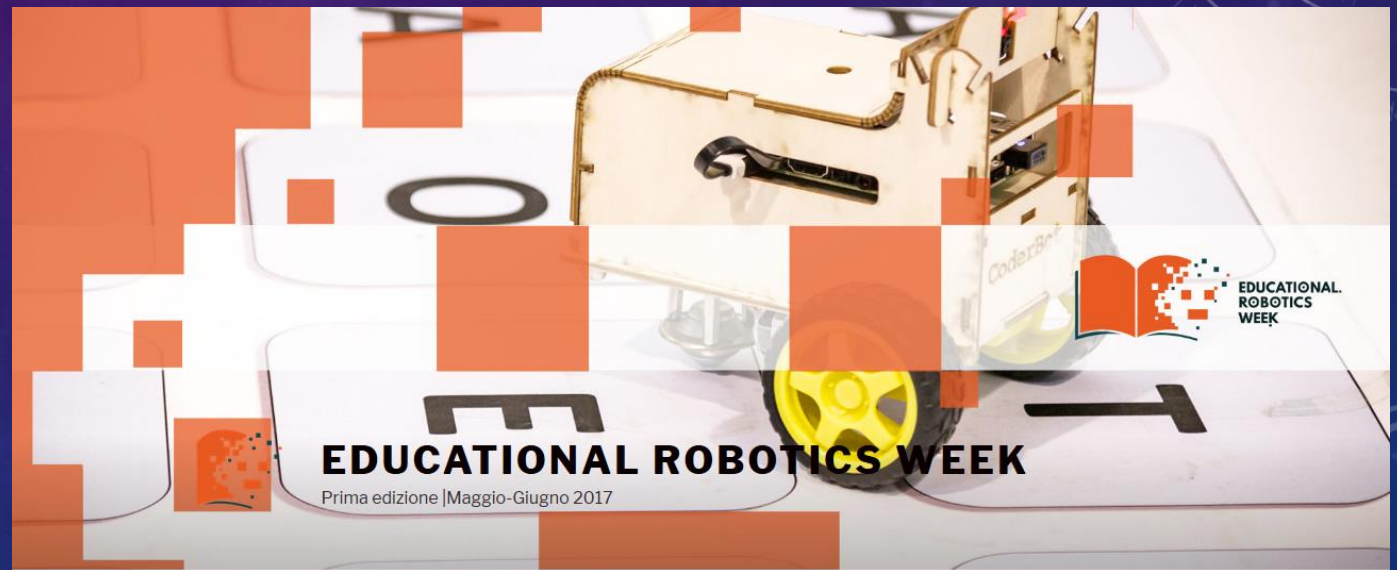
WELCOME

Coding in your Classroom, Now!

Educational Robotics Week

[Link al sito](#) dell'iniziativa organizzata dall'Università Bicocca di Milano

[Amico Robot](#) sito che raccoglie le esperienze realizzate da una rete di scuole della Provincia di Milano e Pavia



Rendere la didattica inclusiva e ... più interessante

Clicca sull'immagine per aprire Symbaloo.

Symbaloo è una raccolta di link. Cliccando sulle caselle è possibile aprire i siti internet collegati.



ALCUNI LINK UTILI

[Programma il futuro](#)

[Code.org](#)

[CodeWeek](#)

[Scratch](#)

[PNSD](#)

- [Linea del 20 con Scratch](#)
- [Coding e competenze linguistiche in una classe 2[^] primaria](#)
- [Cody Roby, attività/gioco unplugged](#)
- [Cody Way, attività/gioco unplugged](#)

Scuola dell'infanzia

- [Progetto Codinggioco](#)