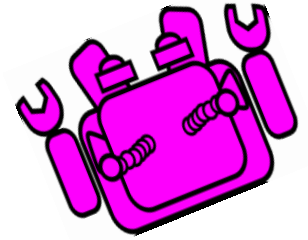
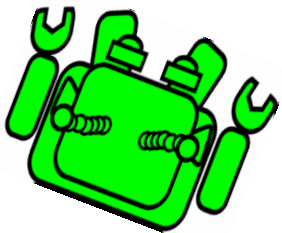


Introduzione al



coding



12 febbraio 2020

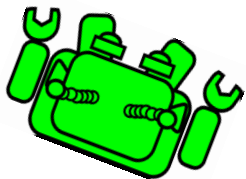
Un approccio informale

<https://www.youtube.com/watch?v=vNaNxwATJqY>

Che cos'è il coding?

Il coding, o pensiero computazionale, è uno strumento metodologico che ci consente di risolvere problemi complessi, frazionandoli in problemi semplici.

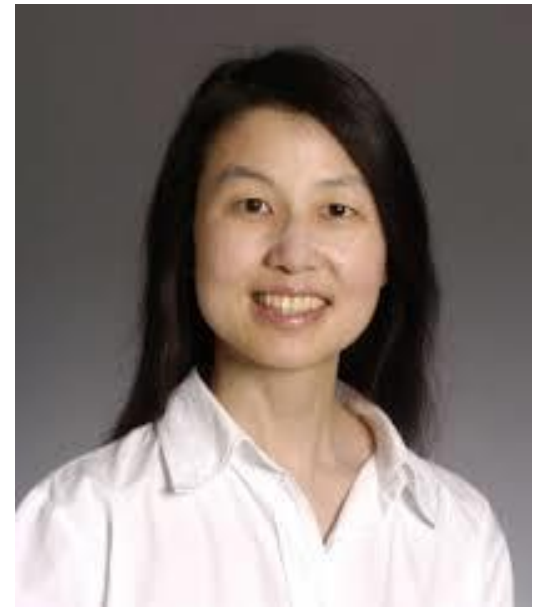
Costruiamo cioè un algoritmo che ci permette di arrivare ad una conclusione certa e ripetibile da tutti, per risolvere lo stesso problema.



<https://goo.gl/cGUC6b>

Che cos'è il pensiero computazionale?

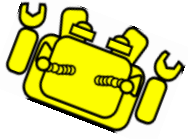
Il pensiero computazionale comporta la risoluzione di problemi, la progettazione di sistemi, la comprensione del comportamento umano attraverso quei concetti tipici solitamente attribuibili al campo della computer science.
(Wing, 2006)



Il coding è per tutti

Quando si diventa fluenti a leggere e scrivere non lo si fa solamente per diventare uno scrittore di professione. **Imparare a leggere e scrivere è utile a tutti.** Ed è la stessa cosa per la programmazione. La maggior parte delle persone non diventerà un esperto di informatica o un programmatore, ma *l'abilità di pensare in modo creativo, pensare schematicamente, lavorare collaborando con gli altri [...]* sono cose che le persone possono usare, indipendentemente dal lavoro che fanno.

(Mitchel Resnick)



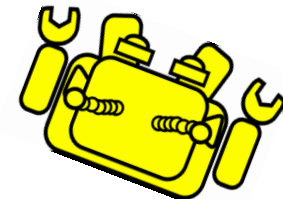
Il coding è per tutti

Il pensiero computazionale è una **capacità trasversale**, utile in qualsiasi ambito di vita, di studio e di lavoro ed è opportuno svilupparla prima possibile.

La scuola è l'ambiente ideale per sviluppare tale capacità, perché è **per tutti**.

Azione #17 del PNSD:

portare il pensiero logico-computazionale a tutta la scuola primaria.



Il pensiero computazionale: le fasi

Gli informatici si trovano spesso a dover progettare soluzioni per cose che altre persone hanno solo sognato - cose che non sono mai state create prima.

Affrontare un problema che non è mai stato risolto può fare paura, ma con questi semplici strumenti, tutto diventa possibile.

Fase 1) Decomporre – Qui non si parla solo di mostri. Stiamo parlando, piuttosto, di suddividere problemi grandi e difficili in qualcosa di molto più semplice. Spesso i grossi problemi altro non sono che tanti piccoli problemi messi insieme.

Fase 2) Schemi – A volte, quando un problema è composto da tanti piccoli pezzetti, noterete che questi pezzetti hanno qualcosa in comune. Se non ce l'hanno, potrebbero comunque avere delle forti somiglianze con i pezzi di un altro problema che è già stato risolto in precedenza. Se riuscite ad trovare queste regolarità, individuare i singoli pezzetti diventerà molto più semplice.

Fase 3) Astrazione – Dopo aver individuato uno schema, è possibile astrarre (ignorare) dai dettagli che differenziano le varie cose ed utilizzare delle tecniche generali per trovare soluzioni che funzionano per più di un problema.

Fase 4) Algoritmi – Quando la soluzione è pronta, è possibile scriverla in modo che possa essere eseguita passo per passo. Ciò rende più facile riuscire ad ottenere i risultati attesi

Indicazioni Nazionali e Nuovi scenari

Strumenti culturali per la cittadinanza

.....

.....

- **5.4 Pensiero computazionale:**

.....Nei contesti attuali, in cui la tecnologia dell'informazione è così pervasiva, la padronanza del coding e del pensiero computazionale possono aiutare le persone a governare le macchine e a comprenderne meglio il funzionamento, senza esserne invece dominati e asserviti in modo acritico.

Il coding a scuola

Con i **bambini** è possibile sviluppare il pensiero computazionale attraverso attività ludiche, intuitive e divertenti che insegnano loro a programmare. Il coding a scuola non si limita alla scrittura del codice, il coding diventa ***strumento di apprendimento***.



<http://programmailfuturo.it/>

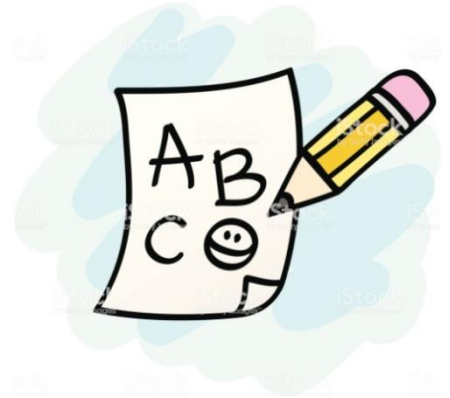
Quali attività?

Le attività che possono essere svolte sono di diverso tipo:

- Lezioni **tradizionali o unplugged**
- Lezioni **tecnologiche**



Lezioni tradizionali



Possono essere svolte senza tecnologie.

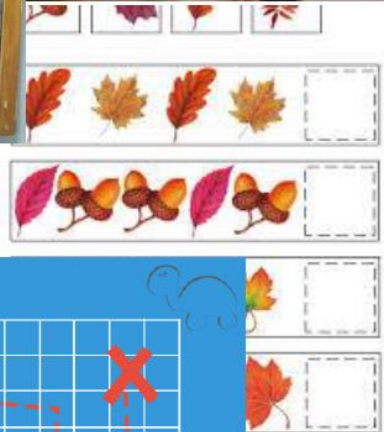
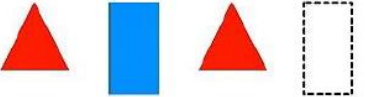

Le strategie **unplugged** (staccato dalla presa elettrica) consentono di proporre attività didattiche dedicate al pensiero computazionale senza fare uso di dispositivi digitali.

<http://programmailfuturo.it/come/lezioni-tradizionali/introduzione>

Lezioni tradizionali

Carta e Matita ???

Complete the patterns
Look at the patterns below and then use the space to the right to complete the pattern. Trace the shape to complete the pattern.

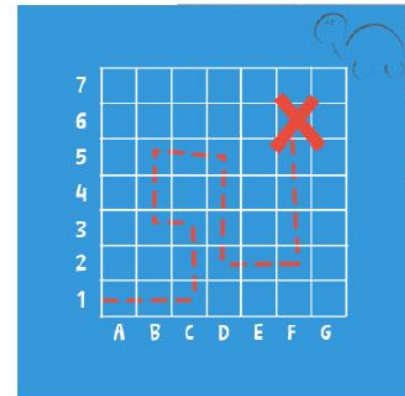


ABSTACTION

Coding Carta e Matita



ALGORITHMIC THINKING



Lezioni tradizionali

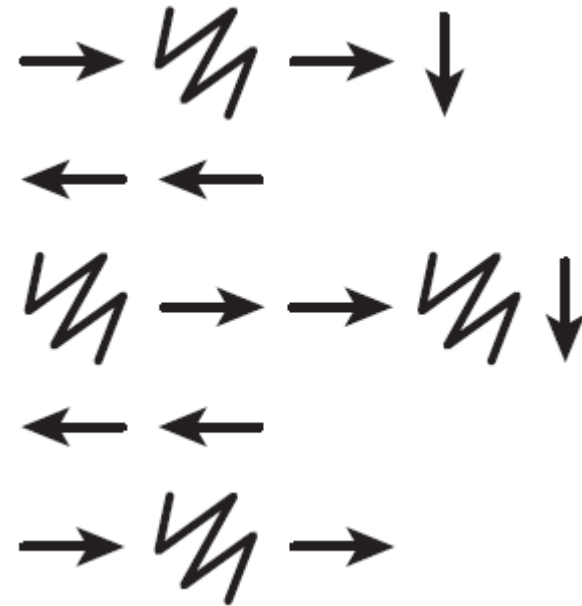
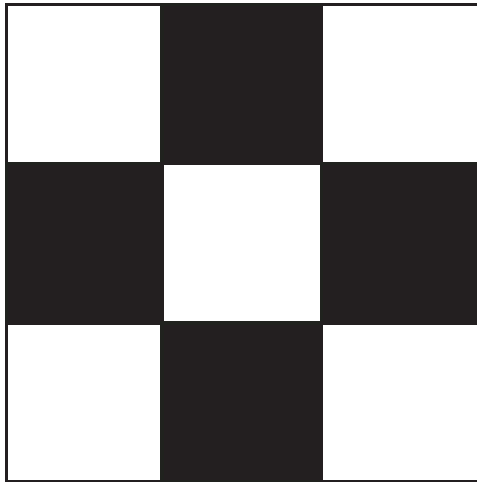
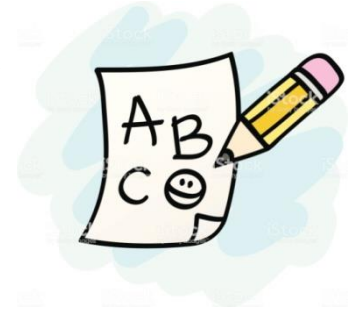


- **Programmazione carta e quadretti**

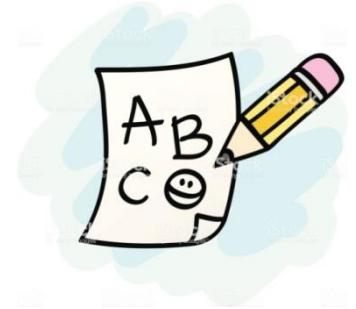
Video d'esempio <https://goo.gl/9gWRaZ>



Lezioni tradizionali



Lezioni tradizionali



PROGRAMMAZIONE SU CARTA E QUADRETTI

FASI:

- 1) Scegli un'immagine dall'insieme.
- 2) Scrivi l'algoritmo per disegnare quell'immagine.
- 3) Converti l'algoritmo in un programma utilizzando i simboli.
- 4) Scambia i programmi con un altro gruppo e disegna la loro immagine.
- 5) Aggiungi delle "funzioni" per rendere più semplici i programmi.
- 6) Scrivi dei programmi per immagini più complesse.
- 7) Scambia i tuoi programmi complessi e disegna nuovamente.

Lezioni tradizionali



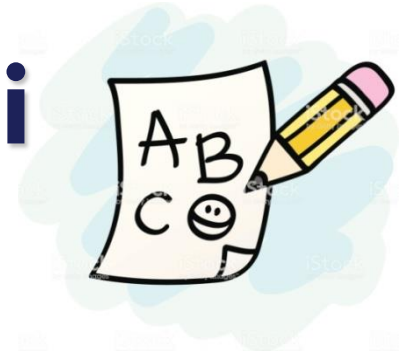
- Programmazione carta e quadretti

Materiale stampabile

<https://programmaitfuturo.it/come/lezioni-tradizionali/programmazione-su-carta-a-quadretti>



Lezioni tradizionali

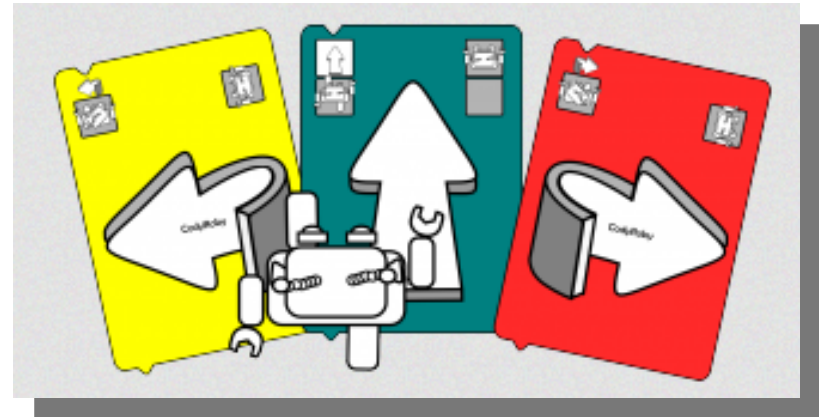


- **Cody Roby**

<http://codeweek.it/cody-roby/>

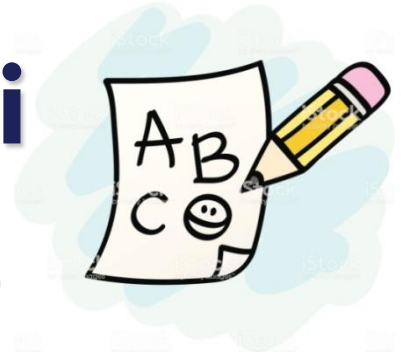
Video d'esempio

<https://goo.gl/5KvTgo>



Lezioni tradizionali

- Il coding alla Scuola dell'infanzia



Video d'esempio

<https://www.youtube.com/watch?v=fRwNOdH6ImE>

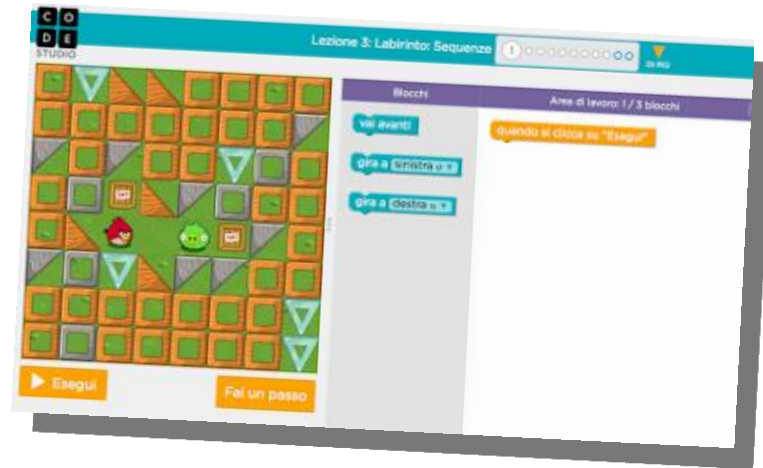
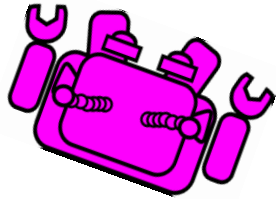
Esempi di percorsi

<https://goo.gl/0VCqXs>



Lezioni tecnologiche

- Sono fruibili tramite web e prevedono una serie di esercizi progressivi per difficoltà. Sono divertenti e coinvolgenti. Permettono di imparare insieme.



<http://programmaitfuturo.it/come/lezioni-tecnologiche/introduzione>

The logo for Hour of Code, featuring the text "HOUR OF CODE" in white on a teal square background, set against a colorful, abstract background of purple, blue, and green.

HOUR
OF
CODE

L'ORA DEL CODICE

CODE WEEK

(5/20 ottobre 2019)

Tutti gli anni si organizza in tutto il mondo una settimana in cui si svolgono attività di avviamento al pensiero computazionale

<http://codeweek.it/>

Per partecipare si può scegliere una o più attività, tradizionali e/o tecnologiche, da svolgere in classe.

HOUR
OF
CODE

L'ORA DEL CODICE CODE WEEK



I NUMERI BINARI

Il sistema numerico binario è un sistema numerico che utilizza solo due simboli, di solito indicati con 0 e 1, invece delle dieci cifre utilizzate dal sistema numerico decimale.

Ciascuno dei numeri espressi nel sistema numerico binario è definito "numero binario".

Il **sistema binario** è il **sistema di rappresentazione numerica** maggiormente utilizzato nel **campo dell'informatica**.

Esso si basa sulla rappresentazione in base due dei numeri che normalmente utilizziamo.

Di conseguenza le **uniche** due cifre utilizzate sono lo 0 e l'1, con cui tramite particolari meccanismi aritmetici è possibile eseguire tutte le operazioni conosciute del **sistema metrico decimale**.

I NUMERI BINARI

Noi usiamo il sistema decimale per eseguire calcoli utilizzando i simboli che vanno da 0 a 9,

i computer invece per calcolare usano il sistema binario, costituito da soli due simboli: **0 e 1.**

È stato scelto così perché i computer capiscono solo lo 0 (non c'è corrente) e l'1 (c'è corrente)

Grazie all'uso di queste due cifre, è possibile rappresentare non solo numeri, ma anche immagini, video, suoni e qualsiasi informazione digitale.

Il computer infatti, prima converte tutti i video, parole, immagini nel sistema decimale e poi esegue i calcoli e una volta convertiti i risultati dal sistema binario a quello decimale, restituisce quindi le immagini, video, audio, e qualsiasi altro tipo di informazione.

I NUMERI BINARI

PROGRAMMAZIONE A BLOCCHI



<https://scratch.mit.edu/>

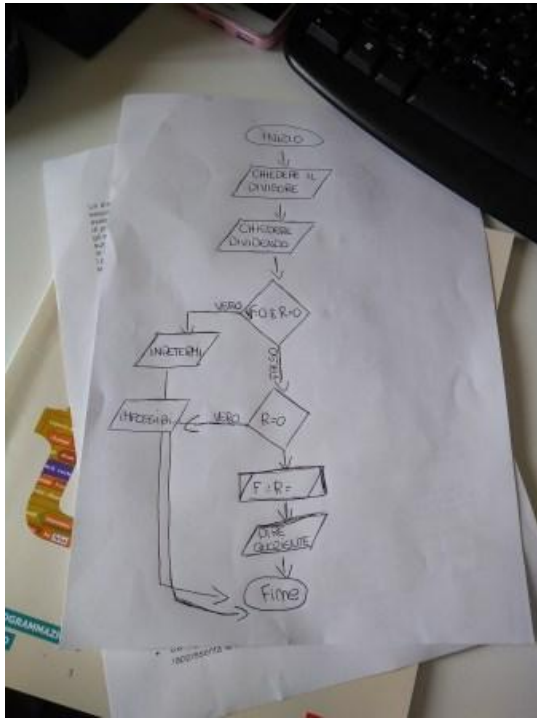
The image shows a screenshot of the Scratch 2 Offline Editor. The main stage displays a penguin sprite in a desert landscape with cacti and rock formations. The penguin is currently at coordinates x: 240, y: -180. The Sprites panel shows the penguin as 'Sprite1'. The Scripts panel is open, showing a script that starts with 'when green flag clicked', followed by 'point in direction 90 degrees', a 'forever' loop containing 'play sound meow', 'move 10 steps', and an 'if touching edge?' block with a 'turn 15 degrees' block inside. The Scripts panel also lists various other blocks like 'move 10 steps', 'turn 15 degrees', 'point in direction 90 degrees', 'point towards', 'go to x: -109 y: -18', 'go to mouse-pointer', 'glide 1 secs to x: -109 y: -18', 'change x by 10', 'set x to 0', 'change y by 10', 'set y to 0', 'if on edge, bounce', and 'set rotation style left-right'. The top of the window shows the title 'Scratch 2 Offline Editor' and system information like 'Mon 2:41 PM' and 'Thinkspace'.

- **PROGETTI PERSONALIZZABILI**
PROGETTI CONDIVISIBILI

DAI BLOCCHI ALLE STRINGHE



<https://smallbasic-publicwebsite.azurewebsites.net/>



```

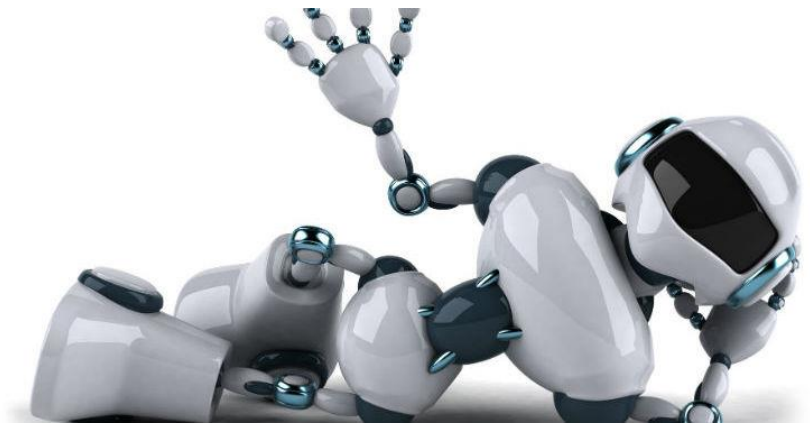
quando si clicca su
  dire "Lo sai che so fare le divisioni?" per 2 secondi
  chiedi "Scrivi il dividendo." e attendi
  porta a a risposta
  chiedi "Divisore?" e attendi
  porta b a risposta
  se a = 0 e b = 0 allora
    porta c a indeterminato
  altrimenti
    se b = 0 allora
      porta c a impossibile
    altrimenti
      porta c a a / b
  fine
  dire "il quoziente è "
  ferma tutto
  
```

```

Senza titolo *
1 START:
2 TextWindow.WriteLine(" DIMMI IL DIVIDENDO ")
3 PRIMONUMERO = TextWindow.Read()
4 TextWindow.WriteLine(" DIMMI IL DIVISORE ")
5 SECONDONUMERO = TextWindow.Read()
6 If (PRIMONUMERO = 0 And SECONDONUMERO =0) Then
7   TextWindow.WriteLine("LA DIVISIONE E' INDETERMINATA")
8 Else
9   If (SECONDONUMERO = 0) Then
10    TextWindow.WriteLine(" NO LA DIVISIONE E' IMPOSSIBILE ")
11   Else
12    QUOZIENTE = PRIMONUMERO / SECONDONUMERO
13    TextWindow.WriteLine(" QUOZIENTE E' " +QUOZIENTE)
14   EndIf
15 EndIf
16 Goto START
17
  
```

Robotica educativa

La **robotica educativa** è l'occasione per l'acquisizione di competenze trasversali, come la capacità di risolvere i problemi (*pensiero computazionale*), lo sviluppo di facoltà cognitive e sociali, la capacità di pianificazione, lo sviluppo dello spirito critico, lo sviluppo della personalità e dell'autostima.



Un robot ...per amico



DASH



BEE-BOT

Un robot ...per amico

Bee-bot



Questo robot è in grado di memorizzare una serie di comandi base (avanti, indietro, ruota a destra e ruota a sinistra) e di muoversi su un percorso in base ai comandi registrati.