

☐ Équipe
☐ Formativa
☐ Piemonte

Pensiero computazionale



Anna Nervo - Andrea Piccione
Webinar - Riconessioni - 9 Marzo 2020



Principio: definizione del PC

- Il pensiero computazionale può essere definito come la risposta alla domanda “come posso fare perché un *agente* risolva questo problema?”, dove *l’agente* potrebbe essere una macchina, una persona o una combinazione di questi.
- Il coding oggi è parte del processo di alfabetizzazione ed è di supporto a competenze chiave del nuovo millennio. (Digicomp 2.1: 1.3-3.4)
- La promozione del pensiero computazionale non è necessariamente legata ad attività di coding.

Principio: PC e competenze

- **Le competenze associate al pensiero computazionale sono diverse a seconda degli autori che ne propongono una definizione:**
 - problem-solving;
 - raccogliere, analizzare e rappresentare dati;
 - utilizzare algoritmi e procedure;
 - realizzare simulazioni;
 - usare modelli.

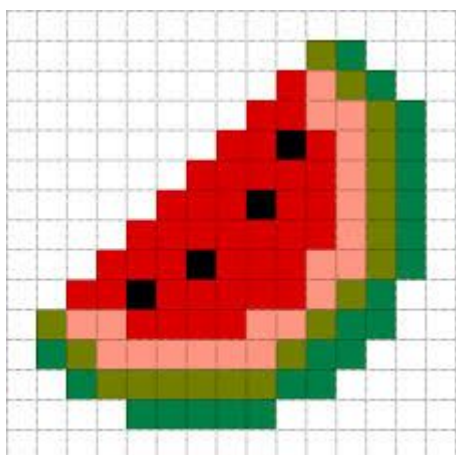
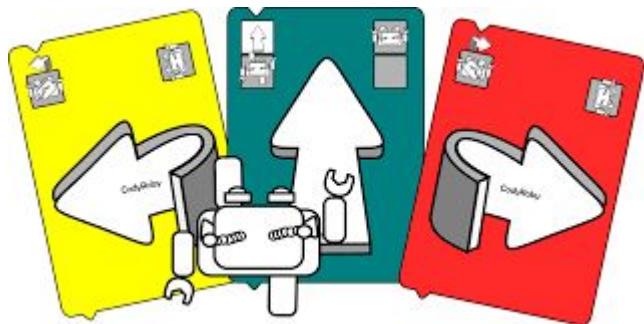
Principio: PC e inclusione

- **Le attività associate al pensiero computazionale sono:**
 - eliminare elementi ridondanti;
 - chiarire le informazioni in ingresso e in uscita;
 - utilizzare un linguaggio semplice ed essenziale;
 - realizzare procedure efficienti;
- **Questo approccio facilita l'inclusione perché aiuta a ridurre e gestire il carico cognitivo degli allievi.**

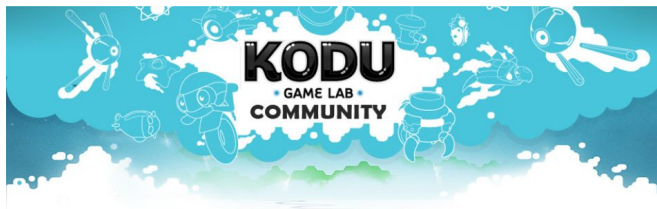
Cosa facciamo: EFT Piemonte

- **In generale: accompagnamento e supporto alle istituzioni scolastiche, ai docenti e agli studenti per la promozione, la sperimentazione e lo sviluppo di metodologie didattiche innovative in relazione al PNSD.**
- **In particolare: supporto, accompagnamento, formazione e assistenza per la didattica a distanza** (Nota 278, 6 Marzo 2020)
- **In questo contributo: proposte per la promozione del pensiero computazionale per il primo e il secondo ciclo con attività a distanza.**

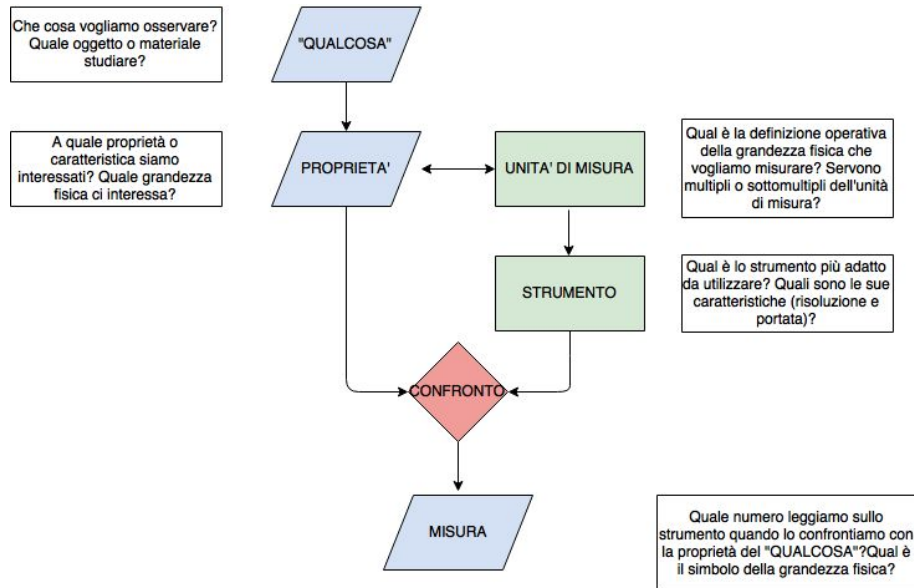
PC unplugged e primi passi



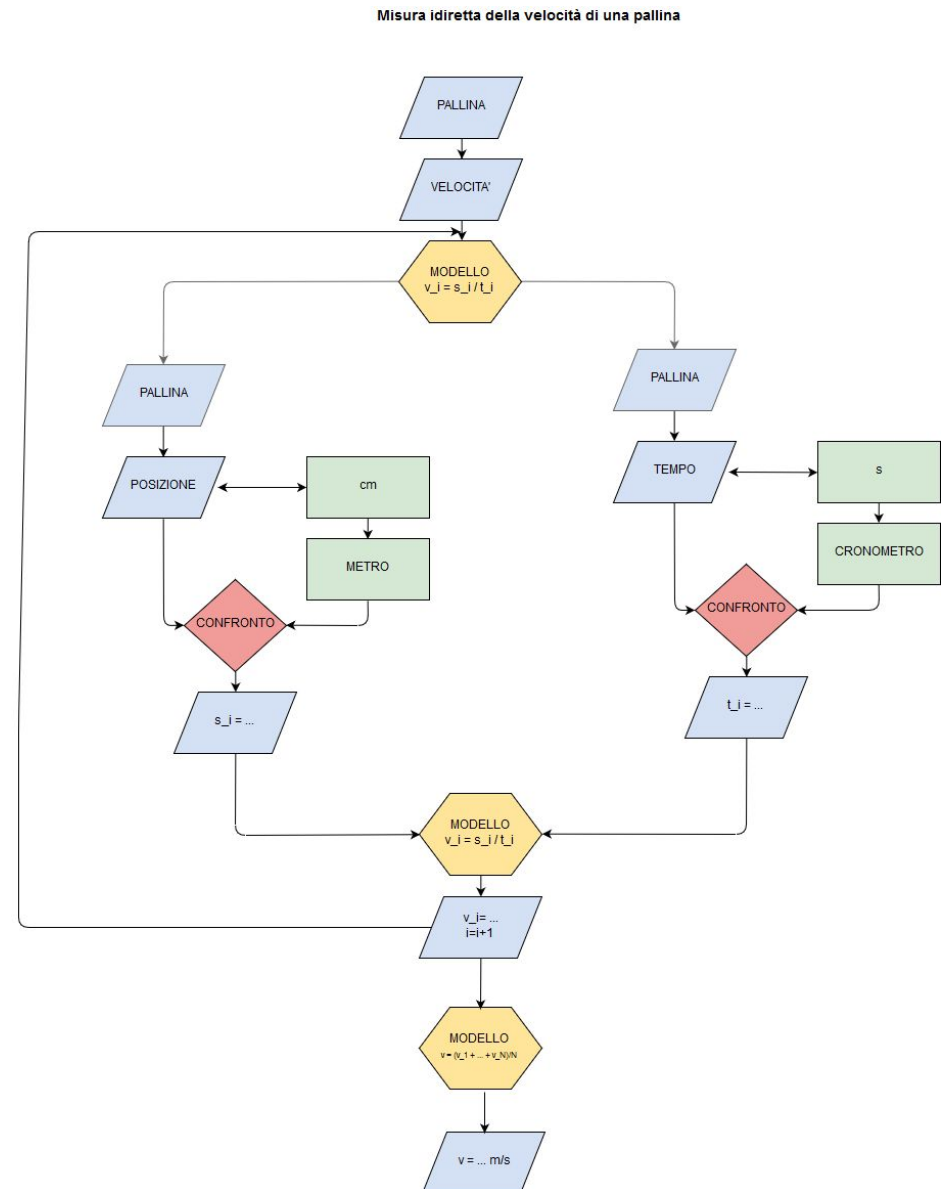
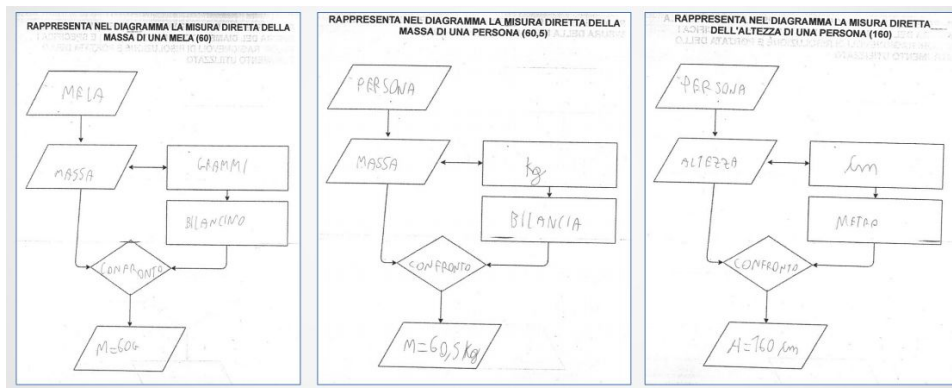
Coding per gioco



Pensiero computazionale e misura



<https://www.draw.io/>



micro:bit e misura

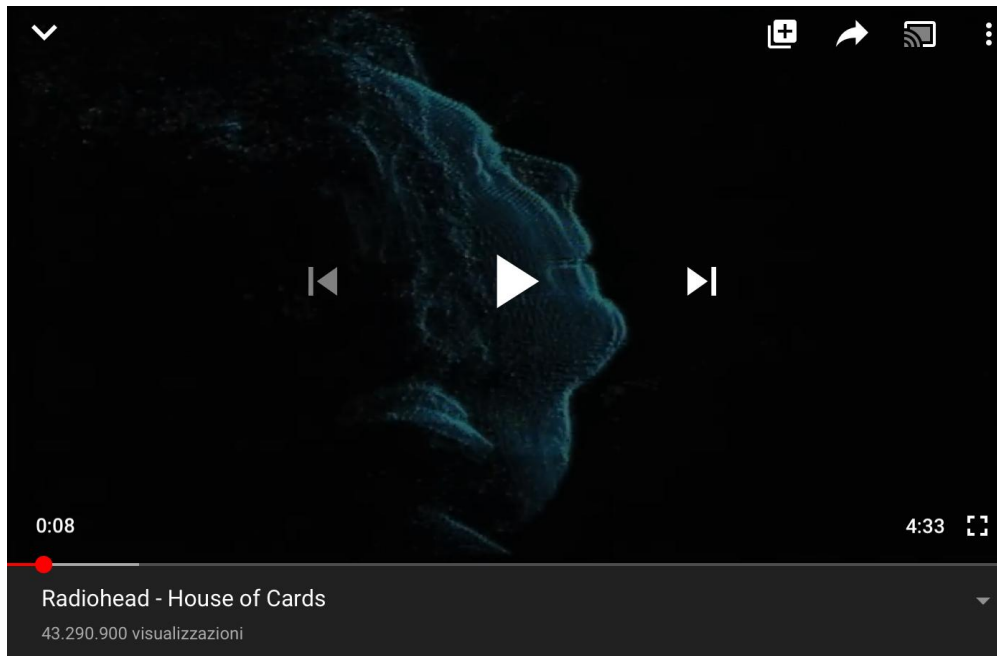
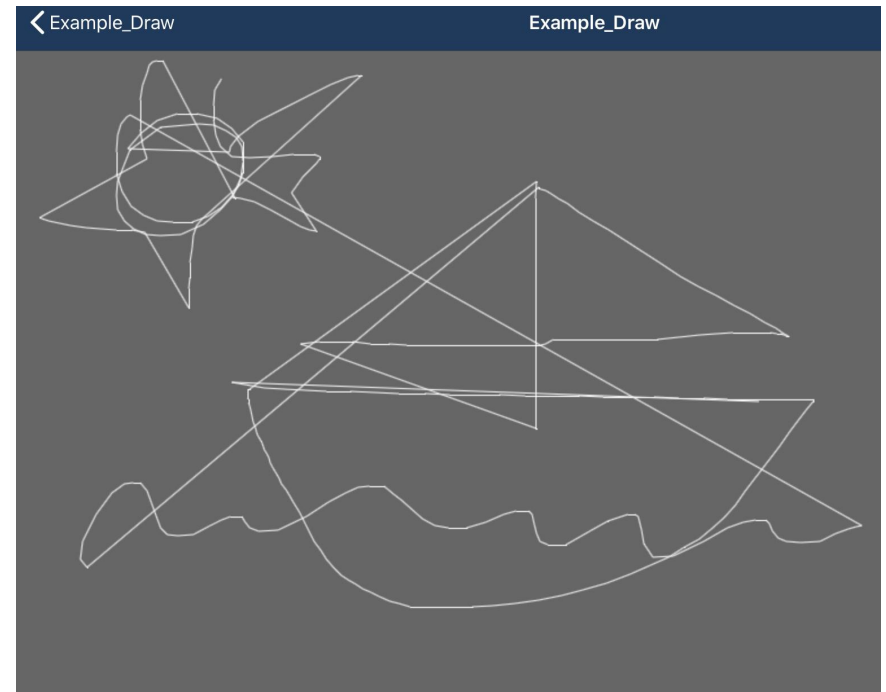
The screenshot displays the Microsoft MakeCode editor interface. At the top, a blue navigation bar includes links for 'Pagina principale', 'Blocchi', 'JavaScript', and the Microsoft logo. On the left, a vertical sidebar lists various block categories: Fondamentali, Ingressi, Musica, LED, Radio, Cicli, Logica, Variabili (highlighted in red), Matematica, and Avanzati. The main workspace is divided into three panes. The left pane shows a digital representation of a micro:bit with pins labeled 0, 1, 2, 3V, and GND. The middle pane, titled 'Variabili', contains a 'Crea una variabile...' button and two 'Prova' blocks: 'imposta Prova a 0' and 'cambia Prova di 1'. The right pane, titled 'Light Level Meter', provides instructions: 'Use this program to graph the light level. Press A to scroll the value on the screen.' Below the text is a code block starting with a 'per sempre' loop. Inside the loop, the code sets a 'reading' variable to 'intensità luce', draws a bar chart of 'reading' up to 255, and checks if the 'pulsante A' is pressed. If pressed, it displays the 'reading' value on the screen. At the bottom of the editor, a purple 'Scarica' button is on the left, a file name 'Light Level Meter' with a save icon is in the center, and a set of control buttons (undo, redo, zoom in, zoom out) is on the right.

Processing

```
< My Projects Example_Draw + ▶
Example_Draw.pde

void setup() {
  size(screen.width, screen.height);
  background(102);
}

void draw() {
  stroke(255);
  if (mousePressed == true) {
    line(mouseX, mouseY, pmouseX, pmouseY);
  }
}
```



Breve corso di Python

SOLOLEARN[COURSES](#)[CODE PLAYGROUND](#)[DISCUSS](#)[TOP LEARNERS](#)[BLOG](#)[SIGN IN](#)

Sign In

Email

Password

☐ Remember me? [Forgot Password?](#)

[SIGN UP](#) [SIGN IN](#)

[f](#) Sign in with Facebook [g+](#) Sign in with Google

GET THE FREE APP

GET IT ON
 Google play

Download on the
 App Store

Learn on the Web

[<](#) **Basic Concepts**

What is Python? 1/13 2 questions	Your First Program 2/13 2 questions	Simple Operations 3/13 4 questions	Floats 4/13 2 questions	Other Numerical Operations 5/13 2 questions
Strings 6/13 3 questions	Simple Input & Output 7/13 2 questions	String Operations 8/13 3 questions	Type Conversion 9/13 2 questions	Variables 10/13 4 questions
In-Place Operators 11/13 2 questions	Using an Editor 12/13 1 questions	Module 1 Quiz 13/13 5 questions		

Python 3 Tutorial

Basic Concepts
1/13

Control Structures
0/13

Functions & Modules
0/9

Exceptions & Files
0/10

Learn

Play

[Take a Shortcut](#)

Riferimenti:

. Link:

- <https://sites.google.com/istruzioneepiemonte.it/eft/home>
- <https://programmailfuturo.it/>

. Articoli:

- Lodi, M., Martini, S., & Nardelli, E. (2017). Abbiamo davvero bisogno del pensiero computazionale?, *Mondo digitale*, Novembre 2017
- Piccione A. (2016), La didattica della misura dall'epistemologia al pensiero computazionale. *Form@re*, 16 (1)
- Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. *In Proceedings of the 2012 annual meeting of the American educational research association*, Vancouver, Canada (Vol. 1, p. 25).
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wing, J.M. (2008). Computational thinking and thinking about computing. *Philosophical transactions of the royal society of London A: mathematical, physical and engineering sciences*, 366(1881), 3717–3725.