

Corso Generazione Web

Sede Corso: IC di Via Anna Botto - Vigevano

Dirigente: *Alberto Panzarasa*

Tutor: *Claudia Verminetti*

Robotica creativa Il pensiero computazionale



**Formazione
docenti**
a.s. 2017/2018

**TERZO
INCONTRO**

Pensiero computazionale: cenni storici

Il concetto di “**pensiero computazionale**” è stato introdotto per la prima volta da **Seymour Papert** nel 1996 parlando di LOGO, il linguaggio di programmazione da lui sviluppato al MIT (Massachusetts Institute of Technology) per insegnare la programmazione ai bambini.



Nel suo famoso libro Mindstorms scrive che la programmazione sviluppa l'intelligenza sequenziale e il pensiero procedurale, insegna come scomporre il problema in componenti più semplici e a trovare eventuali errori se il procedimento non funziona.



Pensiero computazionale: definizione

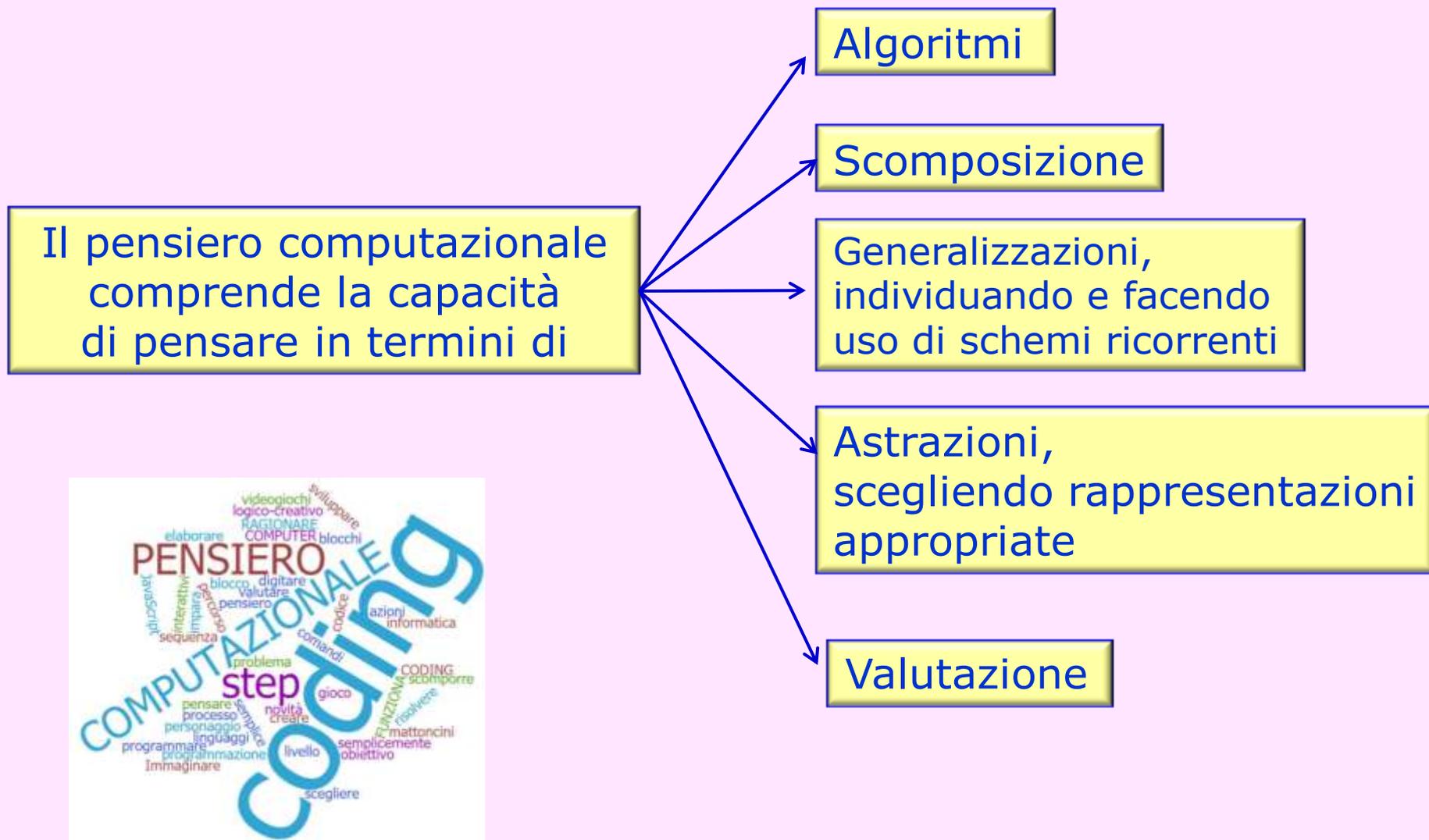
La scienziata informatica Jeannette Wing ha reso popolare l'idea di Seymour Papert sostenendo che il pensiero computazionale dovrebbe far parte delle competenze di tutti i nuovi studenti universitari (Wing, 2006).



Inoltre ha definito il **pensiero computazionale** come:

- ❑ *"...I processi mentali coinvolti nel formulare problemi e le loro soluzioni in modo che le soluzioni possano essere rappresentate in una forma che può essere efficacemente eseguita da un agente di elaborazione dell'informazione (Cuny, Snyder, Wing, 2010, citato in Wing, 2011, p.20).*
- ❑ *La soluzione può essere eseguita da un essere umano o da una macchina, o, più in generale, da combinazioni di uomini e macchine."* (Wing, 2011, p. 20).

Concetti del pensiero computazionale



Concetti del pensiero computazionale: algoritmi

L' **algoritmo** è un insieme finito di operazioni, che devono essere eseguite per portare a termine un dato compito e risolvere un problema.

Il pensiero algoritmico :



- è un modo di arrivare a una soluzione attraverso una chiara definizione dei passaggi
- entra in gioco quando problemi analoghi devono essere risolti più e più volte. Essi non devono essere analizzati ex novo ogni volta, è necessaria una soluzione che funzioni ogni volta.

Un esempio sono gli algoritmi per fare la moltiplicazione o la divisione che si imparano a scuola.

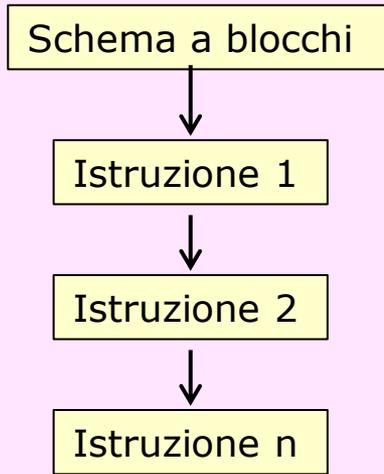
Se le regole sono eseguite con precisione, da un computer o da una persona, si può trovare il risultato di ogni moltiplicazione.

Una volta che ci si è impadroniti **dell'algoritmo**, questo non deve essere elaborato da zero per ogni nuovo problema.

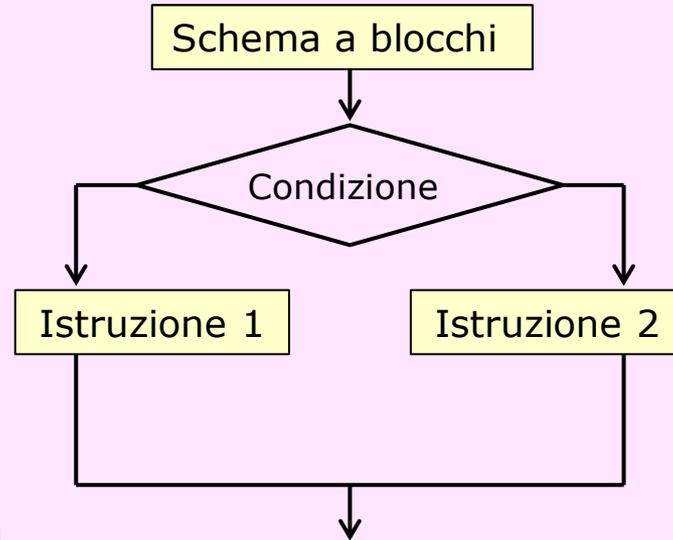
Concetti del pensiero computazionale: algoritmi

È stato dimostrato che qualunque algoritmo si basa su tre strutture fondamentali:

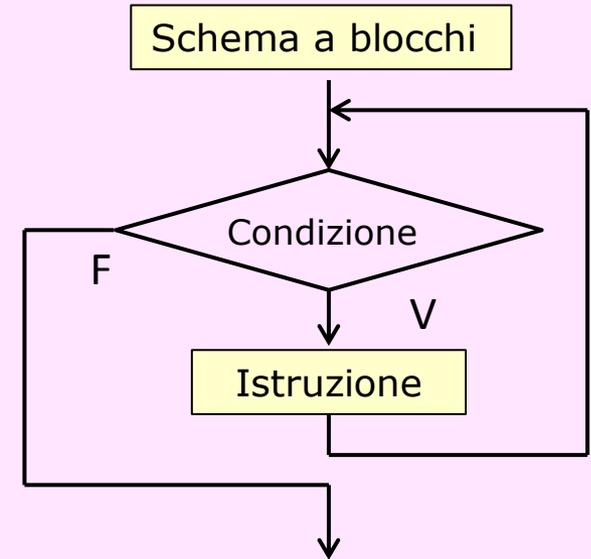
Struttura Sequenziale



Struttura di Selezione



Struttura Iterativa



Concetti del pensiero computazionale: scomposizione

La scomposizione è un modo di pensare ad artefatti in termini delle loro componenti.

Le singole parti possono essere comprese, risolte, sviluppate e valutate separatamente. Questo approccio rende più facile risolvere problemi complessi, permette di comprendere meglio situazioni nuove e facilita la progettazione di grandi sistemi.

Ad esempio:

- Preparare la colazione può essere scomposta in attività separate come: fare un toast; preparare il tè; far bollire delle uova; ecc. Ognuno di queste, a sua volta, potrebbe essere ulteriormente suddivisa in una serie di passaggi.
- Lo sviluppo di un gioco: persone diverse possono progettare e creare i diversi livelli in modo indipendente, dopo aver concordato in anticipo gli aspetti chiave.

Attraverso la scomposizione del compito originale ogni parte può essere sviluppata e integrata più avanti nel processo.

Concetti del pensiero computazionale: generalizzazione

La generalizzazione è associata al saper identificare schemi ricorrenti, somiglianze, connessioni, e a sfruttare queste caratteristiche.

È un modo di risolvere rapidamente problemi nuovi sulla base delle soluzioni di problemi già affrontati, a capitalizzare le precedenti esperienze.

È importante imparare a riconoscere schemi ricorrenti e strategie che vengono utilizzati.

Algoritmi che risolvono alcuni problemi specifici possono essere adattati per risolvere tutta una classe di problemi simili. Quando si incontra un problema di quella classe si può applicare la soluzione generale.



Concetti del pensiero computazionale: astrazione

L'astrazione è il processo di rendere un artefatto più comprensibile attraverso l'eliminazione di dettagli superflui alla sua descrizione.

Un esempio classico è la cartina della metropolitana. La mappa della metropolitana di Milano è un'astrazione con solo le informazioni essenziali per il viaggiatore a pianificare un percorso da una stazione all'altra.

Sono omesse informazioni come la distanza e l'esatta posizione geografica, che risulterebbero inutili.



L'abilità nell'astrazione sta nello scegliere il dettaglio da nascondere in modo che il problema diventi più facile, senza perdere niente di ciò che è importante.

Un aspetto fondamentale di questo processo è nella scelta di una buona rappresentazione del sistema.

Concetti del pensiero computazionale: valutazione

La **valutazione** è il **processo** per **garantire** che una **soluzione**, sia essa un algoritmo, un sistema, o un processo, è **buona**: cioè che sia adatta allo scopo.

Varie **proprietà delle soluzioni** devono essere **valutate**. Sono corrette? Sono abbastanza veloci? Usano le risorse con parsimonia? Sono facili da utilizzare per le persone? Promuovono un'esperienza adeguata?

Nella valutazione, come usata nel pensiero computazionale, c'è una particolare attenzione alla cura del dettaglio.

Raramente vi è un'unica soluzione ideale per tutte le situazioni.



Pensiero computazionale nella vita quotidiana

Le persone usano implicitamente il pensiero computazionale nella vita di tutti i giorni.

Sono chiamate ad esplicitarlo quando devono istruire un soggetto terzo (l'esecutore) a risolvere un problema.

Alcuni esempi

Il ragionamento algoritmico è presente:

- quando organizziamo la nostra giornata
- quando condividiamo i passi di una ricetta che abbiamo sperimentato
- quando facciamo le operazioni aritmetiche
- quando spieghiamo un gioco agli altri
- quando dobbiamo fornire delle istruzioni per raggiungere un luogo
- quando vogliamo costruire una mappa concettuale...



Pensiero computazionale nella vita quotidiana

Il **pensiero computazionale** può tornarci utile in ogni situazione, anche nelle più insolite, come dimostra questo video tratto dal film Apollo 13 e portato come esempio anche dal [sito del progetto](#) [Programma il futuro](#), del Miur, per far comprendere meglio di cosa stiamo parlando.



Pensiero computazionale nella scuola

In Italia, con il documento sulla buona scuola, il governo ha aperto il dibattito sull'introduzione del pensiero computazionale a scuola.



Il dibattito sul documento ha prodotto due proposte di curriculum (una per la scuola dell'obbligo e l'altra per le secondarie superiori) e due documenti di raccomandazioni (uno per l'introduzione dell'informatica nel curriculum e l'altro per l'introduzione del pensiero computazionale come competenza trasversale).

La legge 107 (2015) include il **pensiero computazionale** tra gli obiettivi educativi della scuola (comma 7).

Pensiero computazionale nella scuola

Nella circolare MIUR 08/10/2015 l'introduzione del pensiero computazionale è così motivata:

Il lato scientifico-culturale dell'informatica, definito anche "pensiero computazionale", aiuta a sviluppare competenze logiche e capacità di risolvere problemi in modo creativo ed efficiente, qualità che sono importanti per tutti i futuri cittadini.

Il modo più semplice e divertente di sviluppare il "pensiero computazionale" è attraverso la programmazione (coding) in un contesto di gioco.

In parallelo, il MIUR ha promosso il progetto Programma il Futuro che dall'anno scolastico 2014-2015 sta sperimentando attività di "coding" per l'introduzione del pensiero computazionale a scuola.

Pensiero computazionale e Coding

Le attività di coding possono essere inserite all'interno del Curricolo d'istituto come strumenti per:

- ✓ acquisire competenze trasversali
- ✓ valorizzare le potenzialità di ciascuno
- ✓ potenziare le capacità di attenzione, concentrazione e memoria

L'approccio al coding permette di:

- sperimentare in prima persona
- fare esperienze manuali con materiali che consentono il controllo dell'errore
- provare percorsi per tentativi ed errori cercando anche nuove soluzioni
- vivere l'apprendimento come scoperta
- poter lavorare in autonomia senza l'aiuto dell'adulto
- favorire lo sviluppo di potenziamento della creatività e dei processi logici

Quali strumenti per il coding?

Il **coding** può essere realizzato con **strumenti di programmazione visuale**, sviluppati a scopo didattico e ludico, che offrono la possibilità di sperimentare immediatamente l'effetto delle istruzioni a blocchi composte sullo schermo.

Gli strumenti di programmazione visuale disponibili consentono di risolvere schemi di gioco impartendo istruzioni ad un personaggio o di scrivere veri e propri programmi.

L'effetto delle istruzioni che componiamo è immediatamente visibile e ci consente di imparare dalla nostra stessa esperienza commettendo e correggendo errori.

Il più famoso è Scratch, sviluppato al MIT Media Lab, che offre milioni di progetti condivisi dagli utenti che chiunque può usare, aprire e modificare a piacimento.



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

Quali strumenti per il coding?

È possibile presentare attività “unplugged”.

Ad esempio CodyRoby è un gioco “unplugged” (scollegato) basato sulla programmazione e sull’interpretazione di semplici sequenze di istruzioni elementari.

Cody è un programmatore che impartisce istruzioni, Roby è un robot che le esegue.

Le istruzioni sono carte da gioco, i programmatori (Cody) sono i giocatori, i robot (Roby) sono pedine mosse dai giocatori su una scacchiera, o bambini coinvolti in attività motorie lungo un percorso.

Tutto il materiale è scaricabile da internet.

<http://codeweek.it/lm09-codyroby/>



La prima premessa per lo sviluppo del bambino è la concentrazione. Il bambino che si concentra è immensamente felice.

Maria Montessori