

Il pensiero computazionale

Che cosa significa “pensiero computazionale”?

Nonostante il nome possa sembrare complicato, il **pensiero computazionale** è qualcosa che **usiamo già ogni giorno**, anche senza accorgercene. Si tratta della **capacità di risolvere problemi in modo logico, ordinato e strategico**, proprio come farebbe un computer. Ma attenzione: non significa “pensare come un robot”, bensì **sviluppare un modo di pensare efficace**, che ci aiuta a **trovare soluzioni** passo dopo passo.

Immagina di dover preparare una torta: devi seguire una ricetta precisa, suddivisa in fasi, rispettare l'ordine degli ingredienti e i tempi di cottura. Questo è un esempio di pensiero computazionale nella vita quotidiana.

Perché è utile il pensiero computazionale?

Il pensiero computazionale non serve solo a programmare un computer. È una **abilità trasversale** che può essere applicata in molti ambiti.

- **A scuola:** per risolvere un problema di matematica, scrivere un tema ben organizzato o preparare una presentazione.
- **Nel lavoro:** per pianificare attività, ottimizzare i tempi, prendere decisioni più razionali.
- **Nella vita:** per affrontare situazioni complesse, prendere decisioni informate e risolvere conflitti.

Inoltre, aiuta a **sviluppare il pensiero critico**, la **creatività**, la **capacità di analizzare un problema** e di **trovare soluzioni** innovative.

I 4 pilastri del pensiero computazionale

1. Scomposizione (Decomposition)

Vuol dire **spezzare un problema grande in problemi più piccoli e gestibili**.

Se per esempio devi organizzare una gita scolastica, puoi dividere il compito in fasi diverse, come scegliere la meta, contattare l'autobus, preparare i permessi, organizzare il pranzo, ecc.

2. Astrazione

Significa **concentrarsi solo sugli elementi rilevanti**, tralasciando i dettagli inutili.

Se devi progettare un'app per prenotare i libri in biblioteca, ti interessa sapere titolo e disponibilità, non la marca del computer su cui gira il programma.

3. Riconoscimento di schemi (Pattern recognition)

È la **capacità di identificare modelli o regolarità nei dati**.

Se noti che ogni volta che studi con la mappa concettuale ricordi meglio, hai trovato un pattern utile da applicare.

4. Algoritmi

Un algoritmo è **una serie di istruzioni precise da seguire per risolvere un problema**.

Scrivere le istruzioni passo-passo per accendere un proiettore e collegarlo al PC è un algoritmo!

Esempi concreti in informatica

- **Coding**: scrivere un programma in **Scratch** o **Python** è un'applicazione diretta del pensiero computazionale.
- **Robotica educativa**: usare piccoli robot (come **mBot** o **Lego Mindstorms**) aiuta a trasformare il pensiero in azione concreta.
- **Creazione di videogiochi**: anche realizzare un semplice gioco richiede progettazione logica e suddivisione in compiti.

Esempi concreti nella vita scolastica

- In **matematica**, per risolvere problemi complessi a più fasi.
- In **scienze**, per elaborare dati di un esperimento in modo ordinato.
- In **storia**, per organizzare eventi e cause-effetto in modo cronologico.
- In **educazione civica**, per analizzare le cause di un problema sociale e pianificare soluzioni.

E nella vita di tutti i giorni?

Anche se non ci hai mai pensato, anche nella **vita di tutti i giorni** applichi il **pensiero computazionale**, per esempio per organizzare lo studio in modo più efficiente, per risolvere un conflitto tra amici cercando soluzioni passo passo, oppure per prendere decisioni consapevoli, valutando i pro e i contro.

Il pensiero computazionale **non è solo una competenza per informatici o programmatori**. È una **forma di intelligenza moderna**, che ci permette di affrontare problemi con metodo, di ragionare con chiarezza e di essere più preparati in un mondo sempre più digitale. Coltivarlo a scuola aiuta a essere più autonomi, creativi e responsabili.