

Pensiero Computazionale

Tale concetto è stato introdotto nella legge 107 all' art 1 comma 7. Nella stessa, la parola “coding” o “programmazione”, non è scritta in senso informatico. Ed è giusto che sia così, in quanto potremmo insegnare il pensiero computazionale anche senza programmare!



Ma tipicamente programmare è
un'attività svolta da persone
specializzate che creano programmi
ossia software da fare eseguire ad
un computer!

L'informatica ha per oggetto lo
studio dei fondamenti
dell'informazione e della sua
computazione

Computazione

Il termine computazione deriva dal latino computare che vuol dire calcolare o valutare mediante conteggio diretto o su dati attendibili.

Il computo dunque presuppone conto (o conteggio) e calcolo....

Ma non solo



**Ma vediamo di capire
la differenza tra
contare, calcolare e
computare**

Contare

Significa valutare quantitativamente un insieme, attribuendo numeri in ordine progressivo agli elementi che lo compongono;

Es. contare gli alunni presenti in una classe, ecc.

Calcolare

Significa determinare
attraverso operazioni
aritmetiche.

Es. calcolare una distanza, il
volume, l'area, ecc.

Computare

Significa calcolare o valutare mediante conteggio diretto o su dati attendibili

Es. computare il tempo necessario per raggiungere una certa destinazione

Quindi il computo presuppone anche il calcolo ossia precisi percorsi metodologici da seguire per la risoluzione di problemi (non necessariamente matematici)



I passaggi documentati , che precedono alla scrittura di un programma che poi portano alla fase finale che è la “codifica”, rappresentano quello che viene definito “il pensiero computazionale”.

- Il pensiero computazionale è un insieme di procedure che vengono poi attuate da un esecutore, che opera nell'ambito di un contesto prefissato, per raggiungere degli obiettivi assegnati.
- Il pensiero computazionale è un processo mentale per la risoluzione di problemi ed è costituito dalla combinazione di metodi caratteristici e di strumenti intellettuali, entrambi di valore generale.

Gli strumenti intellettuali includono:

- confidenza nel trattare la complessità
- ostinazione nel lavorare con problemi difficili;
- tolleranza all'ambiguità;
- abilità nel trattare con problemi definiti in modo incompleto;
- abilità nel trattare con aspetti sia umani che tecnologici, in quanto la dimensione umana è essenziale per il successo di qualunque sistema informatico;
- capacità di comunicare e lavorare con gli altri per il raggiungimento di una meta comune o di una soluzione condivisa.



Possiamo affermare che si computa per risolvere (mediante un algoritmo) un determinato problema

Ma diceva Donald Ervin Knuth, informatico statunitense, che se è vero che un problema non si capisce fino in fondo finché non lo si deve insegnare a qualcuno, a maggior ragione quello che si deve insegnare ad una macchina, che è espresso tramite un algoritmo, deve essere compreso in modo ancora più approfondito



Nel mondo odierno i computer sono dovunque e costituiscono un potente strumento di aiuto per le persone. Per essere culturalmente preparato a qualunque lavoro uno studente di adesso vorrà fare da grande è indispensabile quindi una comprensione dei concetti di base dell'informatica. Esattamente com'è accaduto in passato per la matematica, la fisica, la biologia e la chimica.



I nostri alunni cosiddetti nativi digitali utilizzano il Computer o le altre tecnologie disponibili (SmartPhone, Tablet, ecc)e hanno molta esperienza e molta familiarità ad interagire con le nuove tecnologie, ma non a creare usando le nuove tecnologie e neanche ad esprimersi attraverso le nuove tecnologie.

E' come se riuscissero a leggere ma non a scrivere con le nuove tecnologie, in quanto, se così fosse, dovrebbero sapere programmare!



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.

Ma i linguaggi di programmazione come ad es. il C, C++, Java e quant'altro, sono complessi e possiedono una sintassi che richiede un tempo di apprendimento specifico e una età minima di 15-16 anni. Dovremmo aiutare i giovani a diventare **fluente** in modo che possano scrivere con le nuove tecnologie. E questo significa che loro devono essere veramente in grado di descrivere i propri programmi per il computer, o programmare.

Ci dovrebbe essere un linguaggio che permettesse di programmare in modo semplice, intuitivo e scorrevole e che dunque non rappresentasse un ostacolo, ma indicasse in modo naturale al computer le azioni che si vogliono compiere, si potrebbe utilizzare come strumento di insegnamento e non come finalità dell'insegnamento. Per fortuna, questo strumento esiste, si chiama Scratch ed è anche OpenSource.

Tante persone si rendono conto dell'importanza di imparare a programmare anche se si pensa che sia qualcosa che solo un piccolissimo gruppo, con particolari abilità matematiche e una formazione informatica, possa essere in grado di fare. E di sicuro se programmare rimarrà così, saranno solo piccoli gruppi a saperlo fare. Ma programmare non deve necessariamente essere così. Infatti esistono programmi che dopo averli creati si può cliccare su "condividi", e condividere tale progetto con altre persone, in modo che lo possano usare e iniziare a lavorarci.



Inoltre cosa ancora più importante è quella di programmare per imparare perché imparando a programmare, si imparano mille altre cose, aprendo nuove opportunità di apprendimento.

Quando si impara a leggere, allora si potrà anche leggere per imparare. E imparare a programmare è la stessa cosa. Se si impara a programmare, allora si potrà anche programmare per imparare. Quando si impara a programmare, si apre la possibilità di imparare molte altre cose.

Ma, come iniziare col barlume di un'idea e trasformarla in un progetto completamente sviluppato e funzionante?

Basta apprendere i principi basilari della progettazione, sperimentare nuove idee, prendere idee complesse e spezzettarle in parti più semplici, collaborare con altre persone per i propri progetti, trovare e correggere errori nel software quando qualcosa va storto, essere tenace e perseverare nonostante lo scoraggiamento che si prova quando le cose non stanno andando bene.



Queste sono tutte abilità importanti che non hanno a che fare solo con la programmazione di codici. Sono importanti anche per qualsiasi attività.

Imparare a leggere e scrivere è utile a tutti. Ed è la stessa cosa per la programmazione



Magari non si diventerà esperto di informatica o programmatore, ma l'abilità di pensare in modo creativo, pensare schematicamente, lavorare collaborando con gli altri - queste sono abilità che si sviluppano, senza poi pensare che programmare permette anche di esprimere le proprie idee e sentimenti a livello personale.



Gli strumenti disponibili sono di elevata qualità didattica e scientifica, progettati e realizzati in modo da renderli utilizzabili in classe da parte di insegnanti di qualunque materia.



Non è necessaria alcuna particolare abilità tecnica né alcuna preparazione scientifica. Il materiale didattico può essere fruito con successo da tutti i livelli di scuole. E si raccomanda, soprattutto alle scuole primarie, di avvicinare i propri studenti allo sviluppo del pensiero computazionale.

**Esempi didattici concreti; Programmare con scratch ai
seguenti link:**

<https://scratch.mit.edu/projects/115609774/#editor>

<https://scratch.mit.edu/projects/115611205/#editor>