

Corso Generazione Web

Sede Corso: IC di Via Anna Botto - Vigevano

Dirigente: *Alberto Panzarasa*

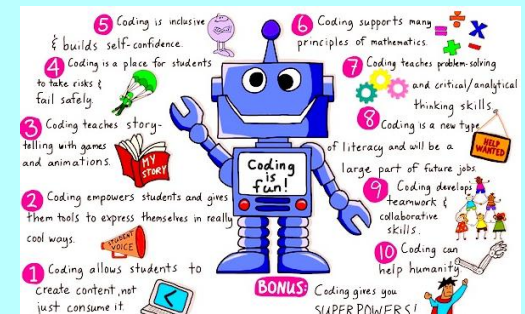
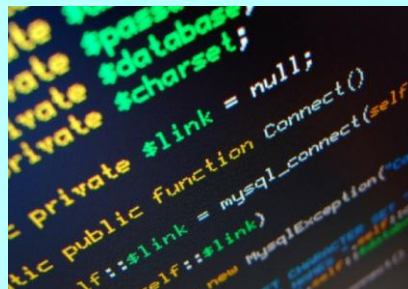
Tutor: *Claudia Verminetti*



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

Robotica Creativa I concetti di base della programmazione, del coding e della robotica

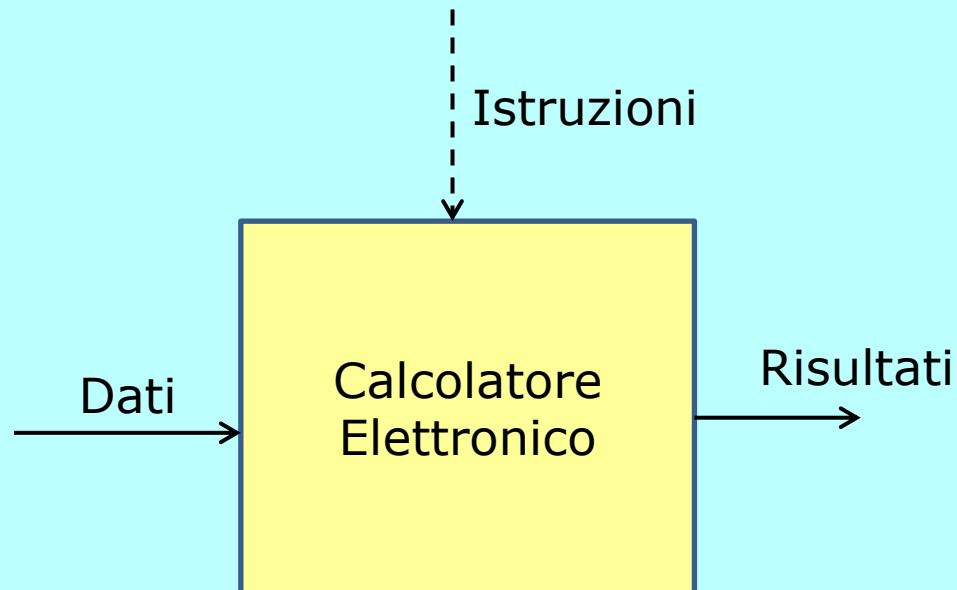
SECONDO INCONTRO



Alcuni Concetti base della programmazione

Con il termine **programmazione** si indica una sequenza logicamente ordinata di operazioni o di istruzioni: l'esecuzione, da parte di un elaboratore, di questo insieme di istruzioni produce la soluzione di un certo problema.

In altre parole, la **programmazione** è l'attività con cui si predispone l'elaboratore ad eseguire un particolare insieme di azioni su particolari informazioni (dati), allo scopo di risolvere un certo problema.



I problemi da risolvere

I problemi che siamo interessati a risolvere con l'elaboratore sono di natura molto varia. Per esempio:

- Somma di due numeri interi
- Calcolare il massimo comun divisore fra due numeri dati
- Dato un insieme di parole, metterle in ordine alfabetico
- Dato un elenco di nomi e relativi numeri di telefono trovare il numero di telefono di una determinata persona
- Dati gli archivi dei dipendenti di un'azienda, calcolare lo stipendio medio del dipendente dell'azienda
- ...

Risoluzione dei problemi

Con questo termine si indica il **processo** che:

- ❑ dato un problema e,
- ❑ individuato un metodo risolutivo



Trasforma i dati iniziali nei corrispondenti risultati finali.

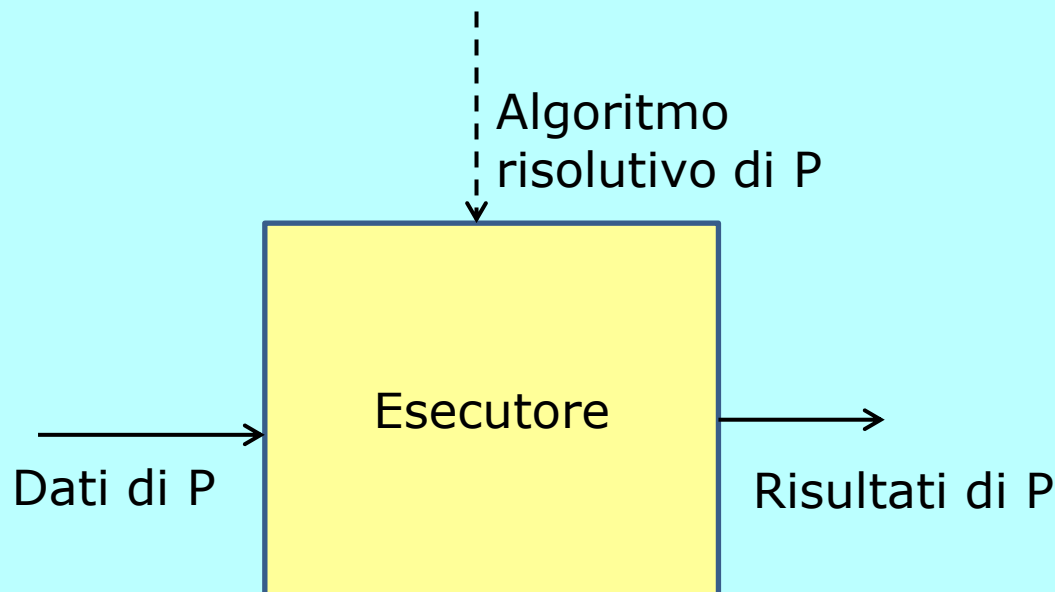
- Affinchè la risoluzione di un problema possa essere realizzata attraverso l'uso del calcolatore, tale processo deve poter essere definito come un insieme di **azioni elementari**, esprimibili mediante **istruzioni**.



Algoritmo

L'insieme ordinato delle azioni che risolve un dato problema P è l'**algoritmo**

- L'algoritmo descrive un metodo risolutivo attraverso un insieme ordinato di azioni
- L'esecuzione dell'algoritmo è affidata ad un generico "**esecutore**", cioè una macchina astratta, non necessariamente un calcolatore, in grado di interpretare ed eseguire ogni azione specificata nell'ordine indicato



Esecutore e istruzioni primitive

- Ad un generico esecutore e` associato un insieme di istruzioni primitive (**set di istruzioni**):
 - sono le sole istruzioni che e` in grado di interpretare ed eseguire.



Esempio: la preparazione del caffè

Esecutore: persona corredata di caffettiera "moka", cucina a gas e macina-caffè;

Algoritmo:

1. **svitare** la caffettiera;
2. se si dispone di caffè macinato:
 - **riempire** il filtro con il caffè macinato,
 - altrimenti se si dispone di caffè in chicchi:
 - * **macinarlo** e ripetere il punto 2;
 - * altrimenti **terminare** (il caffè non si può fare..).
3. **riempire** la parte inferiore della caffettiera con acqua;
4. **inserire** il filtro nella macchina;
5. **avvitare** la caffettiera;
6. **accendere** il fuoco a gas;
7. **collocare** la moka sul fuoco;
8. **attendere** l'uscita del caffè;
9. **spegnere** il fuoco;
10. **fine** (il caffè è pronto).



Esempio: la preparazione del caffè

Esecutore: persona corredata di caffettiera "moka", cucina a gas e macina-caffè;

- Set di istruzioni:**
- **operazioni fondamentali sulla caffettiera:**
 - svitare
 - avvitare
 - riempire il filtro
 - riempire con acqua
 - spostare

 - **operazioni fondamentali sulla cucina a gas:**
 - accendere
 - spegnere

 - **operazioni fondamentali sul macina-caffè:**
 - macinare

 - **altre operazioni:**
 - verifica di condizioni
 - ripetizione di operazioni
 - attesa
 - ...

Proprietà fondamentali dell'Algoritmo

- 1. Eseguitività:** ogni "istruzione" deve essere eseguibile da parte dell'esecutore dell'algoritmo;
- 2. Non Ambiguità:** ogni istruzione deve essere univocamente interpretabile dall'esecutore
- 3. Finitezza:** il numero totale di azioni da eseguire, per ogni insieme di dati di ingresso, è finito

→ Se almeno una delle 3 proprietà non è soddisfatta, la sequenza non è un algoritmo.

Altre proprietà desiderabili:

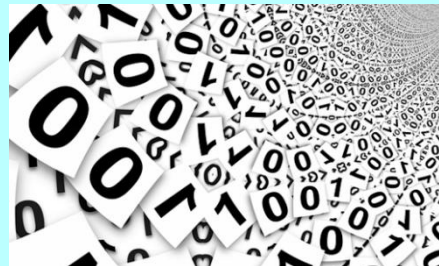
- **generalità:** corretto funzionamento dell'algoritmo anche variando alcuni aspetti del problema (ad esempio, la dimensione dell'insieme dei dati, il tipo dei dati, ecc.)
- **efficienza:** tanto minore è il numero di azioni eseguite per la risoluzione del problema, tanto maggiore è l'efficienza.
- **determinismo:** possibilità di prevedere esattamente prima dell'esecuzione la sequenza di azioni che verranno eseguite, per ogni insieme di dati.
- ...

Algoritmi e Programmi

- Se l'esecutore è un elaboratore elettronico:
 1. È necessario conoscere l'insieme di istruzioni che è in grado di interpretare
 2. È necessario conoscere quali tipi di informazioni (dati) è in grado di rappresentare

Gli aspetti 1. e 2. sono peculiari del formalismo scelto per esprimere l'algoritmo all'interno del sistema di elaborazione, cioè del

Linguaggio di Programmazione



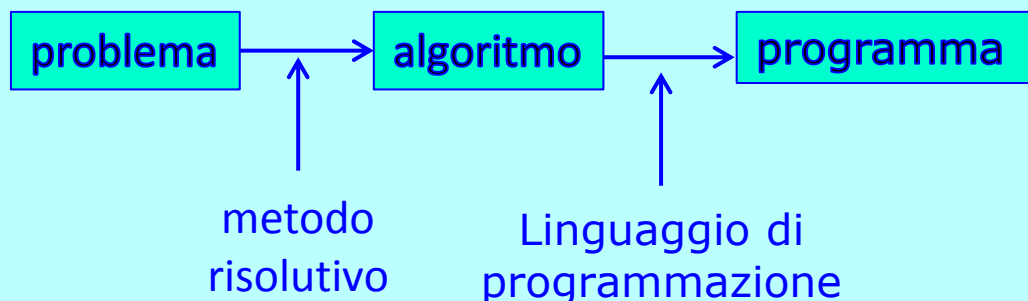
Algoritmi e Programmi

Quindi:

Dato un problema P, la sua soluzione può essere ottenuta mediante l'uso del calcolatore, compiendo i seguenti passi:

1. individuazione di un **metodo risolutivo**
2. scomposizione del procedimento in insieme ordinato di azioni: **algoritmo**
3. rappresentazione dei dati e dell'algoritmo attraverso un formalismo comprensibile per l'elaboratore (il linguaggio di programmazione): **programma**.

Si ottiene così il **PROGRAMMA**, che potrà essere eseguito dall'elaboratore per risolvere automaticamente ogni istanza del problema P.

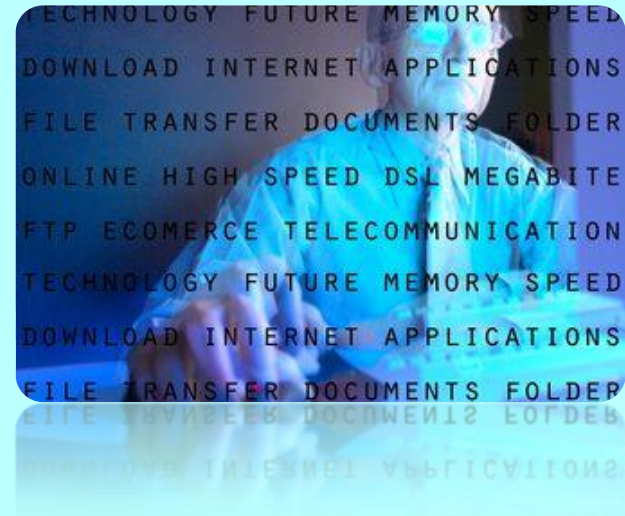


I LINGUAGGI DI PROGRAMMAZIONE

Questi linguaggi permettono al programma di funzionare con la macchina su cui vengono eseguiti, che si tratti di un computer, un telefono cellulare o qualunque altro hardware.

I linguaggi più diffusi al giorno d'oggi sono :

1. **Java!**
2. **C**
3. **C ++**
4. **C#**
5. **Python**
6. **PHP**
7. **Javascript**
8. **Visual Basic.NET**
9. **Delphi/Object Pascal**
10. **Perl**



SCRATCH: una new entry come linguaggio di programmazione utilizzato soprattutto per scopi educativi.

Sembrerebbe che siano stati programmati circa venti milioni di progetti utilizzando il linguaggio **Scratch**.

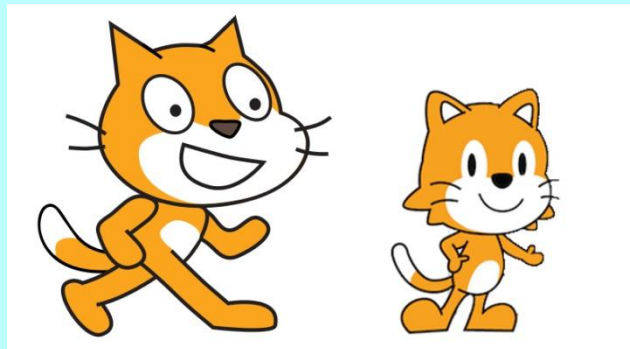
Scratch

Scratch è un linguaggio di programmazione visuale che permette di realizzare:

- storie interattive,
- giochi, animazione,
- musica,
- tanti altri programmi utili a sviluppare la logica e la creatività degli alunni e ad imparare importanti concetti di calcolo matematico e tutto ciò che può essere condiviso con altri gruppi partecipativi..

È disponibile in italiano e permette anche ai docenti di imparare in maniera semplice ed intuitiva a sviluppare supporti multimediali ed interattivi per le loro lezioni, per qualunque disciplina.

Scratch È adatto a partire dai 7/8 anni



Scratch Junior

Programmazione per i più piccoli

È un ambiente di **programmazione a blocchi**, di **programmazione visuale** (senza la necessità di conoscere linguaggi di programmazione), grazie al quale

Anche i bambini di età compresa fra 5 e 8 anni possono programmare storie interattive e giochi. Mentre lo fanno imparano a risolvere problemi, a creare progetti e ad esprimersi in modo creativo usando il computer.

[Scratch Junior](#) istruzioni

[Scratch Jr](#) demo per pc

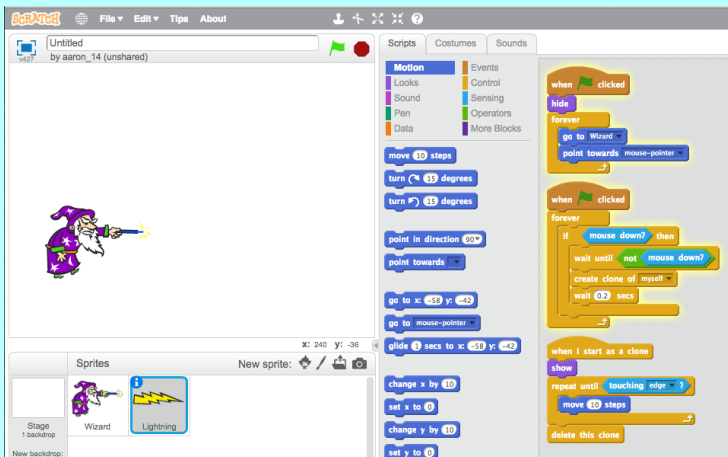


La versione completa è su tablet (Fire, Nexus, ecc.) E iPad. BR: Só uma demo. Una versione completa senza Tablets (Fire (Fogo), Nexus, ecc.) E iPads. "

Scratch

Scratch permette agli studenti di usare da subito, anche senza doverli prima studiare teoricamente, sia tutti i concetti più importanti che sono alla base del coding e del pensiero computazionale che altri più avanzati, come ad esempio:

- Gli Algoritmi
- Le Procedure
- I Cicli
- Le basi degli Oggetti e delle classi
- Le Variabili
- La Programmazione



Consente di ottenere più obiettivi:

- ✓ imparare a programmare,
- ✓ imparare attraverso la programmazione,
- ✓ sviluppare l'abitudine a risolvere problemi più o meno complessi,
- ✓ avvicinarsi in modo giocoso al **coding** e ai principi del pensiero computazionale.

Robotica educativa

La programmazione (coding) è un'attività fondamentale della robotica educativa:

i robot devono seguire le istruzioni impartite attraverso un programma (codice o code)

L'idea rivoluzionaria di far gestire ai bambini i computer e fornire oggetti in movimento che potessero manipolare facilmente e con i quali sperimentare, nacque al MIT di Boston e dalle idee rivoluzionarie di Seymour Papert con il linguaggio LOGO e una prima tartaruga meccanica programmabile

Poi la tartaruga diventa virtuale e infine con Mitchel Resnick si realizzano i primi prototipi di LEGO/LOGO

Nei laboratori del MIT sono state realizzate attività ingegneristiche e di programmazione altamente innovative

Robotica educativa

Da queste esperienze la LEGO Education ha messo in commercio i kit MINDSTORMS (RCX -NXT - EV3) ed è possibile replicare nei laboratori scolastici le esperienze di Papert e Resnick



Tali ambienti sono di norma costituiti da:

Robot + software + materiale curricolare

Aiuta a ragionare su problemi e sistemi, pensando in maniera algoritmica, e consente di imparare a gestire l'errore (prendere consapevolezza dei propri errori per imparare da essi).

Stimola il lavoro in team, la peer-to-peer education, l'autocorrezione degli errori, il coinvolgimento di più sensi, l'inclusione e la condivisione, l'ascolto reciproco e la sana competizione.

Rispetto al coding, aggiunge la dimensione fisica, tangibile e pratica dell'apprendimento, risultando più efficace.

Qualche riferimento normativo

Nella Buona Scuola si parla esplicitamente di introdurre il pensiero computazionale.



L'uso della **programmazione**, della **robotica** e del **gioco** servono per risolvere problemi concreti e far capire, ad esempio, che la Matematica è di fondamentale importanza in tutte le attività umane.

Il **gioco** è una componente che permette di verificare come i concetti matematici tradizionali non sono soltanto teoria, ma strumenti fondamentali nelle applicazioni pratiche.

L'errore non è un tabù ma uno stimolo per ricercare nuove soluzioni (imparare a imparare) e acquisire nuove competenze.

Imparare a programmare deve diventare una risorsa per gli alunni di oggi **per farli diventare non solo utilizzatori attivi dei dispositivi ma soprattutto sviluppatori di nuove idee, di nuovi software.**

Qualche riferimento normativo

Gli elementi che caratterizzano la robotica sono:

- Didattica interdisciplinare
- Apprendimento per scoperta
- Situazioni continue di problem solving
- Attività laboratoriale

la robotica educativa può diventare, grazie al grande **coinvolgimento degli alunni, il modo più semplice per creare un ambiente di apprendimento innovativo, creativo e divertente**



ROBOTICA EDUCATIVA E CODING PER TUTTE LE ETÀ

Scuola dell'infanzia



Scuola primaria



Scuola secondaria



Scuole
Università
Centri di ricerca



Robotica industriale

ROBOTICA EDUCATIVA PER TUTTI

Per introdurre la robotica negli ultimi anni dell'infanzia e nella scuola primaria sarebbe utile far iniziare i bambini con l'ape Bee-Bot / BlueBot

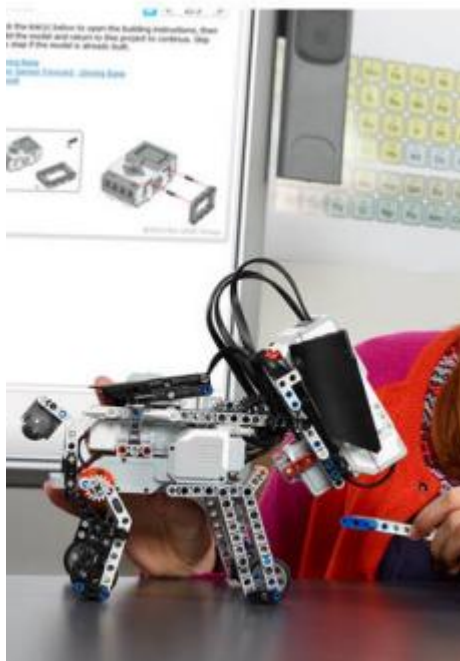
E' la versione reale della "tartaruga" di Papert

Gli alunni dalle terze alle quinte classi della primaria potranno prendere confidenza con la costruzione di oggetti e la loro programmazione con il kit e il software LEGO EducationWeDo e WeDo 2.0



ROBOTICA EDUCATIVA PER TUTTI

Gli studenti degli ultimi anni della scuola primaria e delle scuole secondarie di primo e secondo grado potranno cimentarsi con il più complesso e coinvolgente kit LEGO Education EV3 a vari livelli, con la costruzione e la programmazione dei robot





Il coding è un termine usato per definire la **programmazione informatica**. Applicato al mondo scolastico

Indica l'uso di strumenti e metodi di programmazione visuale a blocchi per favorire lo sviluppo del pensiero computazionale

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





Il MIUR in collaborazione con il CINI, ha avviato il progetto "Programma il Futuro" che rende disponibili alle scuole una serie di lezioni interattive e non, che ogni istituzione scolastica potrà utilizzare compatibilmente con le proprie esigenze e la propria organizzazione didattica.

La "buona scuola" muove i primi passi a partire dal pensiero computazionale.

Lezioni e materiali on line sul sito programmmailfuturo.it.



Formazione



[Coding in your Classroom, Now! - Emma](#)

Corso Mooc online erogato dall 'universita' di Urbino.
Prof Alessandro Bogliolo

Sito di riferimento Codemooc.org

Per imparare il coding

Attività

L'ora del codice e' uno dei principali strumenti di alfabetizzazione orientati al coding . E' stata proposta da Hadi Partovi fondatrice di Code.org durante la Settimana dell'educazione informatica nel 2013. E' un gioco didattico a 20 livelli di complessità che permettono ai bambini di acquisire gli schemi ricorrenti della programmazione visuale a blocchi . (CODE.ORG)



Perché si promuovono queste iniziative?

Perché tra le alfabetizzazioni funzionali (leggere, scrivere, contare..) l'uso dell'informatica e l'uso consapevole di internet e' attualmente la competenza più carente ma anche la più richiesta.

La settimana della programmazione code week

Dal 15 al 23 ottobre 2016 si svolge la settimana europea della programmazione, [Europe Code Week](#), arrivata alla quarta edizione. Nel corso della settimana si svolgeranno migliaia di eventi in ogni parte d'Europa per offrire a giovani e giovanissimi l'opportunità di iniziare a programmare! Non si tratterà di semplici seminari, ma di vere e proprie esperienze di programmazione, che consentiranno anche ai più piccoli di divertirsi in modo costruttivo vedendo le proprie idee prendere forma.



DIVERTIAMOCI A PROGRAMMARE !!!

Per concludere

La programmazione non è il fine,
ma il mezzo per imparare ad
apprendere e sviluppare una forma
mentis utile anche un domani, nel
lavoro e nella vita.

