

LET'S MATH!

di Ubaldo Pernigo e Marco Tarocco

Let's Math! propone un percorso graduale per l'acquisizione delle competenze matematiche, di cittadinanza e pluridisciplinari, con una forte attenzione verso la valutazione e l'autovalutazione.

Competenze digitali - Utilizzare un ambiente di programmazione a blocchi

La realizzazione di codice, utilizzando un linguaggio di programmazione a blocchi, per risolvere problemi permette allo studente di esplicitare il proprio livello di competenza sia nell'utilizzo di un ambiente di programmazione sia nel problem solving applicato.

Le schede mostrate nelle prossime pagine sono tratte dal testo *Let's math!*.

Le attività possono essere assegnate come compito da restituire on line.

Let's math! propone attività di competenze digitali nelle singole unità ma anche una sezione specifica di avvio al pensiero computazionale, che fa uso proprio del linguaggio di programmazione a blocchi.

*Materiale per i partecipanti all'evento **Imparare a pensare come un matematico: attività dentro e fuori dalla classe del 25/03/2021**, parte del progetto **Didattica Digitale Integrata Plus: spunti pratici per le tue lezioni.***



LE MONNIER SCUOLA



La divisibilità


Lezione 1
I multipli e i divisori

- I multipli di un numero
- I divisori di un numero

Lezione 2
Criteri di divisibilità.
Numeri primi

- I criteri di divisibilità
- I numeri primi e i numeri composti

Lezione 3
La scomposizione
in fattori primi

- Il criterio generale di divisibilità

Lezione 4
Il Massimo Comun Divisore

- Problemi con il M.C.D.

Lezione 5
Il minimo comune multiplo

- Problemi con il m.c.m.

NON C'È PROBLEMA
Problem Solving

- Eratostene ed Euclide: strategie antiche per computer efficienti



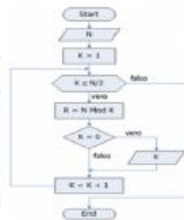
CONTENUTI
DIGITALI

Inquadra il codice per accedere al video e agli audio di questa unità

Ricerca dei divisori di un numero

La ricerca dei divisori di un numero naturale n può essere fatta dividendo il numero dato per i numeri naturali da 1 a n e verificando quali divisioni danno come resto 0. Per i numeri l e n non è necessario fare la divisione perché sono sempre divisori di n . Inoltre, basta fare la divisione per i numeri fino a $n : 2$, perché per i numeri maggiori di $n : 2$ e minori di n la divisione ha sempre quoziente 1 e resto diverso da zero.

È possibile utilizzare un linguaggio di programmazione a blocchi per realizzare una procedura con cui trovare i divisori di un numero qualsiasi. L'algoritmo e il relativo diagramma di flusso sono tradotti nella seguente codifica a blocchi.

**Mettiti alla prova**

- 1 Realizza il programma illustrato utilizzando un ambiente di programmazione a blocchi. Considera degli esempi numerici e verifica i risultati del programma. Gli esempi scelti devono essere inizialmente semplici.
- 2 Introduci un controllo per l'inserimento di un valore 0.
- 3 Inserendo un tempo di attesa di 1 secondo nel controllo "ripeti fino a quando" di quanto si allungano i tempi di esecuzione del codice?

Dare istruzioni: gli algoritmi

Luca è a casa da solo per pranzo per la prima volta e ci ha chiesto di spiegargli come preparare un buon piatto di pasta al dente. Cominciamo!

Istruzioni per cucinare la pasta al dente

1. Prendi una pentola.
2. Riempila metà di acqua.
3. Mettila sul fornello e accendi il fuoco.
4. Aspetta finché l'acqua bolle.
5. Aggiungi un cucchiaino di sale.
6. Versa nella pentola la quantità di pasta che intendi mangiare.
7. Leggi il tempo di cottura sulla scatola della pasta.
8. Un minuto prima che sia trascorso il tempo di cottura assaggia la pasta.
9. Se la pasta non è al dente, attendi qualche secondo e riprova ad assaggiare.
10. Altrimenti scola la pasta.
11. Condisci la pasta con del sugo.



Per dare le istruzioni abbiamo suddiviso le operazioni per cucinare la pasta in una sequenza di azioni più semplici. Abbiamo cercato di essere chiari e di indicare in che ordine eseguire le operazioni.

Anche quando abbiamo davanti un computer e un problema da fargli risolvere, dobbiamo fornirgli gli "ingredienti" necessari, cioè i dati di input, e comunicargli la sequenza esatta di azioni da eseguire per arrivare al risultato.

Il computer, però, non accendono fornelli né riempiono pentole. Eseguono calcoli, leggono informazioni da dispositivi e scrivono informazioni su altri. Un computer legge, per esempio, informazioni da una tastiera e le visualizza sullo schermo, oppure le salva nella sua memoria interna per un utilizzo futuro.

Per svolgere un compito il computer ha bisogno di indicazioni precise, un po' come un cuoco con le ricette: queste indicazioni si chiamano **algoritmi**.

Un **algoritmo** è una sequenza, ordinata e finita, di istruzioni elementari che permettono di arrivare alla soluzione di un problema.

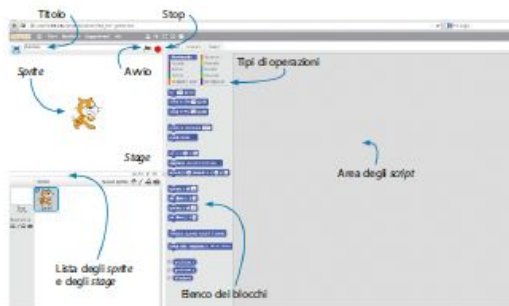
Una sequenza di istruzioni, per essere considerata un algoritmo, non deve avere casi mancanti o ambigui e l'esecutore, colui che esegue le azioni descritte, deve essere in grado di svolgere ogni passaggio. A partire dagli stessi dati, l'algoritmo deve fornire sempre gli stessi risultati, in un tempo finito.

Scriviamo un programma: Scratch

I diagrammi di flusso non risultano comprensibili al computer. L'algoritmo, una volta progettato, va codificato usando un linguaggio di programmazione. Ma non è così semplice utilizzare un linguaggio tanto diverso dal nostro. Per aiutarci usiamo l'ambiente di programmazione fornito da Scratch, che ci permette di comporre un programma usando un linguaggio grafico, a blocchi.

Scratch è un software open source, sviluppato dal Lifelong Kindergarten Group dei Media Lab del MIT, che permette la creazione e la condivisione di storie interattive, giochi e animazioni. La sua particolarità è che consente di comporre programmi senza dover digitare delle istruzioni, ma solo trascinando e componendo i blocchi della sequenza di istruzioni di un algoritmo.

Per scrivere un programma in Scratch basta collegarsi al sito Internet <http://scratch.mit.edu> e passare alla modalità di creazione. Ci troveremo di fronte a una schermata con un'area (area degli script) dove comporre il programma, una finestra (Stage) dove se ne vedrà l'output, e uno spazio centrale da dove "prendere" i blocchi delle istruzioni per poi trascinarli nell'area degli script per comporre il codice.



I programmi Scratch sono associati a personaggi o oggetti, gli *sprite*. Il principale di questi personaggi è un gattino arancione e bianco, che chiameremo Cody. È possibile usare in uno stesso programma più personaggi, o scegliere un personaggio diverso da Cody, cliccando sulla scritta *Nuovo Sprite*.

Lo stage, il palcoscenico sul quale viene mostrato il risultato del programma, è in realtà una griglia: posizionandosi al centro della finestra possiamo leggere poco sotto che ci troviamo nel punto $x: 0, y: 0$. Possiamo sostituire lo sfondo bianco con la griglia xy (ma anche con un fondale marino o un paesaggio creato da noi) cliccando su *Nuovo sfondo*.

Proviamo a scrivere un primo programma in cui Cody si sposta a destra, dice una frase di saluto e poi miagola. Le istruzioni che ci servono sono tutte contenute nella colonna centrale, divise per categorie e contraddistinte da un colore.