

STORTUP

PROGRAMMAZIONE A BLOCCHI

APPRENDERE, COOPERARE, SPERIMENTARE

Edizione OPENSCHOOL

- 1 LIBRODITESTO
- 2 E-BOOK+
- 3 RISORSE**ONLINE**
- 4 PIATTAFORMA

HOEPLI

Stært Up PROGRAMMAZIONE A BLOCCHI Informatica per il primo biennio



EDITORE ULRICO HOEPLI MILANO

Copyright © Ulrico Hoepli Editore S.p.A. 2019 Via Hoepli 5, 20121 Milano (Italy) tel. +39 02 864871 – fax +39 02 8052886 e-mail hoepli@hoepli.it

www.hoepli.it



Tutti i diritti sono riservati a norma di legge e a norma delle convenzioni internazionali

INDICE



PROGETTI CON SCRATCH 3.0

A.1 INTRODUZIONE E PRIMI PROGRAMMI	2	Storytelling digitale	18
Scratch, un linguaggio di programmazione		Storyboard e timeline	18
visuale	3	Editor di immagini	20
Come usare Scratch 3.0	3	Sincronizzare le azioni	23
Gli elementi di Scratch	4	PALESTRA DELLE COMPETENZE	26
• Lo stage	5		
• Sprite e script	6	A.3 DISEGNO, GEOMETRIA E QUIZ	28
Categorie dei blocchi	6	Disegno e geometria	28
• Il sistema di riferimento di Scratch	7	Le variabili	31
La struttura di controllo sequenza	9	Input da tastiera	33
La struttura di controllo ripetizione	11	Timbri e casualità	34
PALESTRA DELLE COMPETENZE	12	I quiz	36
		La struttura di controllo selezione	36
A.2 ANIMAZIONI E STORYTELLING		PALESTRA DELLE COMPETENZE	39
DIGITALE	13		
Gli effetti grafici	13		
Le modalità di animazione	14	A.4 MUSICA E VIDEOGIOCHI	41
Lavorare con i suoni	14	Musica	42
• La scheda Suoni	15	Videogiochi	44
• Duplica, commenta e cancella	15	Arcade	45
Programmazione concorrente	17	Programmazione a eventi	50
Scomposizione di un problema	17	PALESTRA DELLE COMPETENZE	53



PROGRAMMI CON MICRO:BIT

B.1 INTRODUZIONE E PRIMI PROGRAMMI	56	B.2 ANIMAZIONI, MUSICA E GIOCHI	70
Che cos'è l'IoT	57	Animazioni e sprite	71
Che cos'è micro:bit	57	Numeri casuali con micro:bit	72
Come usare Microsoft MakeCode		Programmi musicali con micro:bit	75
for micro:bit	58	Liste e funzioni	77
Primi programmi con micro:bit	59	Giochi	79
• Gestione progetti, script e blocchi	60	Testi e codici	80
Input/output e componenti aggiuntivi	61	• Il codice Morse	80
PALESTRA DELLE COMPETENZE	69	PALESTRA DELLE COMPETENZE	83



FLOWCHART CON FLOWGORITHM

C.1 ALGORITMI, FLOWCHART		C.2 CICLI, MATEMATICA E STATISTICA	98
E PROGRAMMI	86	Struttura di controllo ripetizione	99
Problema, algoritmo, programma	87	Cicli infiniti	100
 Proprietà di un algoritmo 	87	Copiare, tagliare e cancellare blocchi	101
Flowchart e Flowgorithm	88	Cicli pre-condizionati e post-condizionati	102
• Installazione di Flowgorithm	89	Errori di programmazione e debug	103
• Flowchart base e tipi di blocco	89	• <i>Debug</i> in Flowgorithm	103
Algoritmi + Strutture Dati = Programmi	90	Il valore di pi greco	105
Ciclo di vita del software	93	Elementi di statistica	106
Struttura di controllo selezione	94	• Il fattoriale	108
PALESTRA DELLE COMPETENZE	97	PALESTRA DELLE COMPETENZE	110

PRESENTAZIONE

ST@RTUP - PROGRAMMAZIONE A BLOCCHI è un testo destinato agli studenti del **primo biennio** dei diversi indirizzi scolastici per introdurli alla **programmazione informatica** (*coding*).

Il volume unisce logica e creatività con la semplicità dei linguaggi di programmazione visuali basati sull'assemblaggio di blocchi di istruzioni. Il testo è strutturato in tre moduli, ciascuno dedicato a uno specifico strumento/linguaggio (Scratch 3.0, micro:bit e Flowgorithm) e guida lo studente in un percorso che sviluppa il pensiero computazionale e il coding attraverso la costruzione di programmi interessanti. Questo progetto rappresenta una novità nell'insegnamento della programmazione, in quanto mette assieme in uno stesso percorso didattico i tre linguaggi sopra elencati, che apportano un contributo peculiare nell'ambito della logica, degli algoritmi e della programmazione.

L'esposizione dei contenuti è semplice, dettagliata, arricchita di richiami, precisazioni, suggerimenti e corredata da numerosi esercizi; le procedure operative sono affiancate da un congruo numero di immagini esplicative. Lo studente è coinvolto come parte attiva e allenato alla logica, all'ordine procedurale, alla correttezza espositiva, alla sicurezza operativa e all'autonomia progettuale.

L'opera mira a sollecitare negli alunni un approccio alla conoscenza attraverso la sperimentazione, favorendo l'abitudine alla progettazione, al lavoro per obiettivi basato sulla collaborazione e sul confronto.

Il testo **accompagna** lo **studente** in un preciso percorso didattico e costituisce un riferimento chiaro per il suo apprendimento e per l'acquisizione delle competenze previste dalle Linee guida ministeriali.



Contenuti del volume

Modulo A - PROGETTI CON SCRATCH 3.0

Ouesto modulo presenta il linguaggio e l'ambiente di programmazione a blocchi probabilmente più conosciuto e utilizzato al mondo. Vengono presentati i **concetti** base, le strutture di controllo sequenza e ripetizione, effetti grafici, suoni e cambi di immagine per realizzare animazioni e racconti digitali. La struttura di controllo selezione, i blocchi per il disegno e per l'input da tastiera e le variabili consentono di scrivere programmi di geometria animata e quiz.

Nella parte finale, vengono introdotti i **blocchi** per la **musica** e viene applicato quanto visto precedentemente per progettare e **realizzare programmi musicali e videogiochi**.



Modulo B - PROGRAMMI CON MICRO:BIT

I contenuti di questo modulo forniscono conoscenze e tecniche per un utilizzo appropriato e razionale del **simulatore della scheda micro:bit**, mediante l'**ambiente di sviluppo Microsoft MakeCode for micro:bit**. Gli esercizi proposti guidano lo studente a una graduale acquisizione di tecniche che permettono un utilizzo efficace dei software e allenano a individuare soluzioni adeguate alle situazioni proposte.

Il percorso suggerito permette di scrivere **programmi a blocchi** per micro:bit **con ripetizioni**, **variabili**, **input** e **output per la creazione di animazioni**, accendendo e spegnendo i LED del display, semplici **programmi musicali e giochi** con l'uso della casualità.

Vengono infine trattati alcuni concetti sul **codice** e i **testi**, realizzando alcuni programmi che mettono in pratica quanto appreso.



Modulo C - FLOWCHART CON FLOWGORITHM

In questo modulo vengono introdotte una serie di nozioni che consentono di giungere alla determinazione di **algoritmi** e **strutture dati** per la risoluzione di una serie di problemi classici di programmazione, matematica e statistica.

Viene quindi illustrato come disegnare flowchart con Flowgorithm, utilizzando le variabili e le strutture di controllo sequenza, selezione e ripetizione.

Apparato didattico ed esercitativo

Il volume si articola in **tre moduli** tra loro indipendenti che, a loro volta, si compongono di diverse **unità di apprendimento**.

Ogni unità utilizza **icone**, **schemi**, **rubriche** e **impostazioni standard** che guidano lo studente all'**acquisizione di conoscenze** e allo **sviluppo di abilità**.

Le **conoscenze**, la padronanza terminologica specifica, le **abilità** e le **competenze** raggiunte sono irrobustite e verificate puntualmente in ogni unità mediante la proposta di test teorici e prove pratiche, volti a **rafforzare la capacità di ragionamento e di logica** e a sviluppare e **consolidare le competenze digitali**.



Nella versione digitale del volume (eBook+) i test a risposta chiusa della sezione Palestra delle competenze sono fruibili in versione interattiva e autocorrettiva.

Materiali online

I file delle immagini e degli audio da utilizzare nello svolgimento dei Progetti proposti all'interno delle unità e nelle schede Progetto e creo! (sezione Palestra delle competenze) del modulo A sono archiviati nella cartella MODU-LO_A_Scratch_3.0, scaricabile dal sito www.hoepliscuola.it (hoepliscuola.it).

STRUTTURA DEL CORSO PER IMMAGINI

A, PROGETTI CON SCRATCH 3.0 ELISA INTRODUZIONE E PRIMI PROGRAMM ELISA ANDICACIO SI CONTROLLANO DICINAL ELISA ANDICACIONE SI CONTROLLANO DICINAL ELISA ANDICACIONE SI CONTROLLANO DICINAL ELISA ANDICACIONE SI CONTROLLANO DICINAL CONTROLLANO DICINAL CONTROLLANO CONTROLLANO

MAPPA CONCETTUALE

In apertura di ogni unità, la mappa concettuale offre una sintetica anticipazione dei contenuti sviluppati, fungendo al contempo da schema riepilogativo e di sistematizzazione dei saperi per abituare lo studente all'apprendimento e alla sintesi dei contenuti attraverso un percorso ragionato.

PROGETTI, PROGRAMMI, FLOWCHART

Esercizi guidati che accompagnano lo studente nell'apprendimento dei concetti teorici permettendogli di meglio comprendere i contenuti proposti.

CURIOSITÀ

Box di consigli o curiosità sugli argomenti sviluppati e approfondimenti su alcuni contenuti trattati.

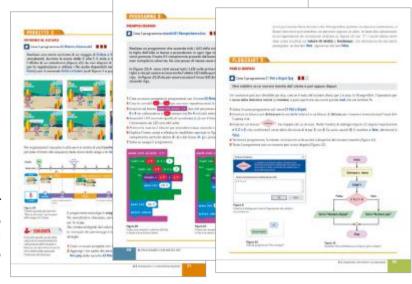
LET'S TALK ABOUT...

Box di informazioni e curiosità in inglese sui contenuti trattati nei paragrafi.

NOTA BENE

Richiama l'attenzione con avvertimenti e consigli riguardanti alcuni contenuti proposti.







PROGETTO E CREO!

Scheda presente al termine di ogni unità, con esercizi che allenano lo studente all'applicazione delle conoscenze, permettendogli di acquisire abilità e competenze.

A DOMANDA... RISPONDO!

Scheda presente al termine di ogni unità: attraverso test di diversa tipologia, consente di verificare l'acquisizione dei contenuti. Nella versione digitale (eBook+) questi test sono interattivi e autocorrettivi.



L'OFFERTA DIDATTICA HOEPLI

L'edizione **Openschool** Hoepli offre a docenti e studenti tutte le potenzialità di Openschool Network (ON), il nuovo sistema integrato di contenuti e servizi per l'apprendimento.

Edizione OPENSCHOOL















LIBRO DI TESTO



eBOOK+



RISORSE ONLINE



PIATTAFORMA DIDATTICA



Il libro di testo è l'elemento cardine dell'offerta formativa. uno strumento didattico agile e completo. utilizzabile autonomamente o in combinazione con il ricco corredo digitale offline e online. Secondo le più recenti indicazioni ministeriali, volume cartaceo e apparati digitali sono integrati in un unico percorso didattico. Le espansioni accessibili attraverso l'eBook+ e i materiali

integrativi disponibili

nel sito dell'editore

sono puntualmente

richiamati nel testo

tramite apposite icone.

L'eBook+ è la versione digitale e interattiva del libro di testo. utilizzabile su tablet, LIM e computer. Aiuta a comprendere e ad approfondire i contenuti, rendendo l'apprendimento più attivo e coinvolgente. Consente di leggere, annotare, sottolineare, effettuare ricerche e accedere direttamente alle numerose risorse digitali integrative.

→ Scaricare l'eBook+ è molto **semplice**. È sufficiente seguire le istruzioni riportate nell'ultima pagina di questo volume. Il sito della casa editrice offre una ricca dotazione di risorse digitali per l'approfondimento e l'aggiornamento.
Nella pagina web dedicata al testo è disponibile MyBookBox, il contenitore virtuale che raccoglie i materiali integrativi che accompagnano l'opera.

Per accedere ai materiali è sufficiente registrarsi al sito

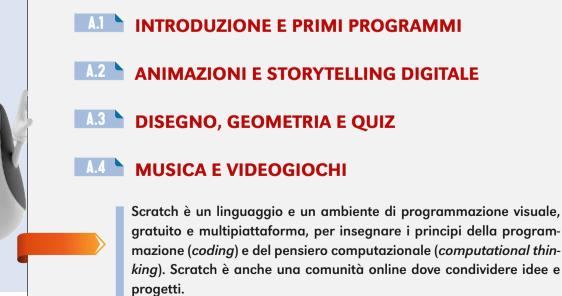
www.hoepliscuola.it
e inserire il codice
coupon che si trova nella
terza pagina di copertina.
Per il docente nel sito
sono previste ulteriori
risorse didattiche
dedicate.

La piattaforma didattica è un ambiente digitale che può essere utilizzato in modo duttile, a misura delle esigenze della classe e deali studenti. Permette in particolare di condividere contenuti ed esercizi e di partecipare a classi virtuali. Ogni attività svolta viene salvata sul cloud e rimane sempre disponibile e aggiornata. La piattaforma consente inoltre di consultare la versione online deali eBook+ presenti nella propria libreria.

→ È possibile accedere alla piattaforma attraverso il sito www.hoepliscuola.it.

PROGETTI CON SCRATCH 3.0





A. PROGETTI CON SCRATCH 3.0

INTRODUZIONE E PRIMI PROGRAMMI

A.2 ANIMAZIONI E STORYTELLING DIGITALE

A.3 DISEGNO, GEOMETRIA E QUIZ

A.4 MUSICA E VIDEOGIOCHI

OBIETTIVI UNITÀ

- Conoscere i concetti base di Scratch.
- Conoscere le strutture di controllo sequenza e ripetizione.
- Progettare e realizzare con Scratch semplici animazioni.





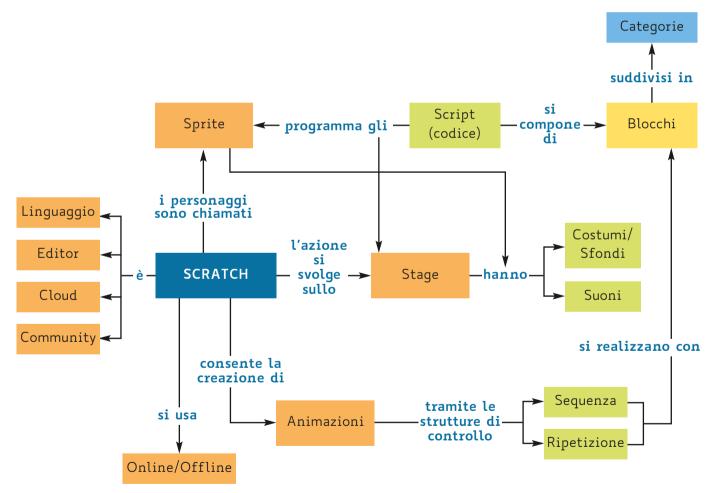




Figura 1 Logo di Scratch

Scratch è uno degli ambienti più utilizzati al mondo per imparare a programmare.

La **versione 3.0** ha reso ancora più bello ed efficace questo fantastico linguaggio a blocchi, che unisce logica, tecnica e fantasia.



Figura 2Mitchel Resnick, papà di Scratch e direttore del Lifelong Kindergarten Group ai Media Lab del MIT

Figura 3 Scratch è un ambiente di condivisione di progetti di programmazione creativa



Scratch, un linguaggio di programmazione visuale

Scratch è un linguaggio e un ambiente di programmazione visuale, gratuito e multipiattaforma, per insegnare i principi della programmazione (coding) e del pensiero computazionale (computational thinking). Scratch è anche una comunità online dove condividere idee e progetti.

Scratch è nato nel 2007 al Lifelong Kindergarten Group dei Media Lab del **MIT** (*Massachusetts Institute of Technology*) **di Boston** e si rivolge in particolare alla fascia d'età 8-16 anni.

Possiamo usare Scratch per realizzare in maniera semplice e divertente **animazioni, fumetti e storie interattive, simulazioni, quiz e giochi**, mixando tra loro **contenuti multimediali** di varia natura: immagini, testi, suoni e animazioni. Scratch si comporta un po' come il DJ di un tempo, che mixava la musica muovendo e rallentando ad arte i piatti del giradischi, in modo tale che i dischi in vinile venissero "graffiati" (to scratch, graffiare) dalle testine di lettura.

Il motto di Scratch è: «Immagina, programma, condividi» e attorno al progetto, la cui mascotte è lo Scratch Cat, è nata una vasta comunità online, con più di 50 milioni di progetti condivisi. Scratch non è solo un linguaggio di programmazione a blocchi semplice e intuitivo, ma è anche un editor che offre un ambiente di programmazione per creare programmi. Inoltre, Scratch mette a disposizione una piattaforma cloud per salvare e condividere i progetti in Internet (Figura 3).

NOTA BENE

Le versioni Scratch 1.4 e 2.0 sono ancora disponibili per il download (https://scratch.mit.edu/download). In Scratch 3.0 è possibile importare programmi realizzati nella versione 2.0 ma non viceversa. I progetti online salvati nei propri account e creati con Scratch 2.0 sono presenti e in generale funzionano anche con la nuova versione.

Figura 4Schermata finale della procedura di registrazione



Come usare Scratch 3.0

Scratch 3.0, rilasciato il 2 gennaio 2019, può essere eseguito sia su **PC** in browser moderni che supportano la tecnologia WebGL, quali Chrome, Firefox, Edge e Safari, sia su **tablet** (con qualche limitazione). Pur mantenendo le stesse modalità di funzionamento della versione 2.0, l'interfaccia si è arricchita di nuovi e interessanti blocchi ed è stata resa ancora più semplice e moderna.

Scratch è utilizzabile sia **online**, collegandosi al sito del progetto, sia **offline**, scaricando e installando sul proprio computer la versione di **Scratch Desktop** per il proprio sistema operativo, disponibile all'indirizzo https://scratch.mit.edu/download.

Noi faremo **riferimento alla versione online**, anche se quello che vedremo funziona pressoché allo stesso modo nelle due modalità.

Possiamo usare Scratch online anche senza registrarci. In questo caso, creiamo una **cartella** (per esempio **scratch**) dove **salvare** i file **.sb3** dei nostri progetti.

Se si vuole anche usufruire del servizio **cloud** di Scratch ed entrare a far parte della sua **community** (la comunità online degli utenti di Scratch) è necessario creare un **account personale**. Per registrarsi basta accedere a https://scratch.mit.edu, fare clic su **Unisciti alla comunità di Scratch**, compilare i campi **username**, **password** e indirizzo **email** e fornire alcune informazioni utili a fini statistici.

Terminata la procedura di registrazione (*Figura 4*), si riceve una **mail** per attivare la possibilità di **condividere** i propri progetti e **commentare** e seguire quelli degli altri.

NOTA BENE

La bandierina verde permette di eseguire uno script all'avvio del programma. Si tratta di un blocco "a cappello" che può essere collocato solo in cima a uno script. E, infatti, non ha alcun incastro per inserire nessun altro blocco al di sopra.

Gli elementi di Scratch

Scratch utilizza la **metafora** dei **giochi di costruzione** e dei **puzzle**. L'idea base è quella di prendere dei pezzi e incastrarli insieme per costruire qualcosa di nuovo e più complesso, nel nostro caso un programma.

Un'altra **metafora** per capire Scratch è quella del **teatro**: sul palcoscenico (**stage**) gli attori (**sprite**) eseguono la parte prevista dal loro copione (**script**), istruzione (**block**) dopo istruzione. A questo punto mancano solo il **regista** e lo **sceneggiatore**: e questi sono proprio i **ruoli** affidati a chi programma con Scratch. Inoltre, programmando, potrà capitare di avere anche i ruoli di costumista, fonico ecc.

Quando si esegue Scratch, si apre in automatico un **nuovo progetto** che contiene uno sprite con un costume dello Scratch Cat. In *Figura 5* è in esecuzione **il nostro primo script**, con due blocchi per lo sprite del gatto.

La lettura e l'**esecuzione** dei blocchi di uno script è **sequenziale dall'alto verso** il basso. Dopo che è stato eseguito l'ultimo blocco, lo script termina.

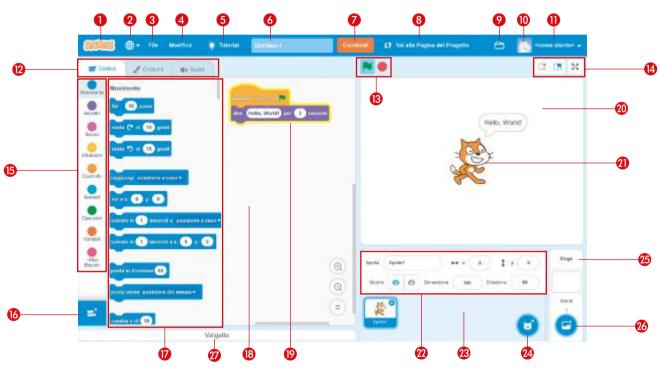


Figura 5 L'editor di Scratch 3.0 mentre sta eseguendo il progetto "Hello World!". Alcuni elementi sono disponibili solo online e se si è effettuato l'accesso col proprio account

- 1. Logo di Scratch (link alla Home Page)
- 2. Scelta della lingua dell'interfaccia
- Menu File con le voci per creare un nuovo progetto, salvare il progetto corrente o salvarne una copia nel cloud, salvare il progetto corrente sul proprio computer o caricarne uno dal computer
- Menu Modifica con le voci per annullare (Ctrl + Z)/ripristinare (Ctrl + Shift + Z) l'ultima azione e attivare/disattivare la modalità turbo, che esegue il programma più velocemente
- 5. Link a tutorial di Scratch
- 6. Nome del progetto
- 7. Condivisione del progetto
- 8. Link alla pagina del progetto
- Link alla pagina "Le mie cose" con l'elenco dei propri progetti

- 10. Immagine dell'utente corrente
- 11. Link al profilo dell'utente corrente
- 12. Schede per Codice (Script), Costumi/Sfondi e Suoni dello sprite corrente o dello stage
- 13. Avvia (Vai) e arresta (Ferma tutto) l'esecuzione del programma
- 14. Pulsanti per diminuire/ingrandire l'area dello stage (per avere più/meno posto per gli script) e per visualizzare lo stage a pieno schermo
- 15. Categorie (ed estensioni) dei blocchi con evidenziata la categoria selezionata
- 16. Pulsante per aggiungere un'estensione
- 17. Elenco dei blocchi della categoria o estensione selezionata
- 18. Area degli script dello sprite corrente o dello stage
- Script in esecuzione (bordo giallo) composto da due blocchi

- 20. Area dello stage, dove appaiono gli sprite
- 21. Sprite di default con un costume dello Scratch
- 22. Pannello di controllo per impostare le proprietà dello sprite corrente
- Pannello degli sprite con la miniatura dello sprite corrente evidenziata dal contorno azzurro. Il clic sull'icona "x" cancella lo sprite.
- 24. Pulsante per creare un nuovo sprite con un certo costume scegliendolo dalla libreria (o galleria), disegnandolo nell'editor di Scratch, pescandolo "a sorpresa" dalla libreria o da un file di un'immagine presente nel computer
- 25. Pannello per la gestione dello stage
- 26. Pulsante per scegliere un nuovo sfondo con le stesse modalità viste al *punto 24* per gli sprite
- 27. Valigetta per condividere e riutilizzare script, sprite, costumi e sfondi tra i propri progetti

Per **modificare un elemento** è necessario prima **selezionarlo**. Per lo stage, si fa clic sulla sua miniatura in basso a destra, mentre per gli sprite è possibile fare clic sulle relative miniature, oppure fare doppio clic sugli sprite stessi nell'area dello stage. Nell'area delle miniature l'elemento selezionato, con cui si sta operando, viene evidenziato con un bordo azzurro.

Prima di analizzare più in dettaglio gli elementi principali di Scratch, realizziamo il nostro primo programma.

PROGETTO 1

HELLO, WORLD!



Realizza il programma composto dai due blocchi di *Figura 5*, che mostra un output (in questo caso un fumetto che dura 2 secondi) con la scritta "Hello, World!". In informatica, quando si affronta un nuovo linguaggio di programmazione, è tradizione partire con questo programma in onore dello scomparso Dennis Ritchie (*Figura 6*).

- 1 Collegati al sito https://scratch.mit.edu e fai clic sul pulsante Crea per aprire l'editor con un nuovo progetto. In alternativa, esegui Scratch installato sul tuo computer, che automaticamente crea un nuovo progetto.
- **2** Se la **lingua dell'interfaccia** non è **l'italiano**, fai clic sul pulsante e seleziona **ltaliano** dalla lista delle lingue disponibili.
- **3** Se hai effettuato l'accesso col tuo account, nella casella per il **nome del progetto** (*Figura 5, punto 6*) digita **A1-Hello World**.
- 4 Nella barra di sinistra (*Figura 5, punto 15*) seleziona la categoria e trascina (*drag and drop*) nell'area degli script il **blocco** "a cappello"



Figura 6
Ken Thompson (a sinistra) e
Dennis Ritchie (a destra),
creatori di Unix, da cui
derivano macOS, GNU/Linux
e Android. Ritchie ha anche
sviluppato il linguaggio C

- **5** Seleziona la categoria, trascina il blocco dire Cisol per 2 secondi, incastrandolo sotto al blocco precedente e modifica il testo di default in Hello, World! (Figura 5).
- 6 Se hai effettuato l'accesso col tuo account, il programma viene salvato automaticamente nel cloud di Scratch.
- 7 Per avere una copia del programma sul tuo computer, dal menu File scegli la voce Salva sul tuo computer. Se hai operato con Chrome o Firefox, sposta il file scaricato, eventualmente rinominandolo A1-Hello World.sb3, dalla cartella Download alla cartella che hai creato per i programmi Scratch; operando con Edge potrai scegliere la cartella di destinazione durante il salvataggio (d'ora in avanti non lo specificheremo più).
- 8 Esegui il programma facendo clic sulla bandierina verde (Figura 5, punto 13), oppure su uno dei blocchi dello script.

Lo stage

Sullo **stage** si svolge l'azione. I pulsanti (vedi Figura 5, punto 14) consentono di avere l'area per gli script normale (pulsante centrale) o più ampia e di passare alla **modalità presentazione** (o player) a pieno schermo.

Per lavorare con lo stage abbiamo a disposizione:

- una raccolta di **script** (il copione può essere suddiviso in più parti);
- una galleria con uno o più sfondi scenografici (c'è sempre uno e un solo sfondo corrente);
- una collezione di **suoni** (sound).

Anche lo stage ha uno o più script. Infatti, in Scratch come a teatro, dietro le quinte del palcoscenico ci sono uno o più addetti alla scenografia con un copione specifico che indica, per esempio, quando mettere un fondale, oppure far partire una musica di sottofondo.

CURIOSITÀ

La parola **sprite** ("folletto") è legata alla *computer graphics* e, in particolare, ai primi giochi 2D, chiamati *arcade*, i quali presentano immagini bidimensionali, chiamate appunto sprite, che si muovono sullo sfondo del gioco.



Figura 7
Game over su una vecchia
macchina arcade, le antenate delle
moderne console per videogame

NOTA BENE

Attenzione a **non confondere sprite con costume!** Uno sprite, quando è in scena, indossa un solo costume per volta, indipendentemente da quanti costumi sono presenti nel suo quardaroba.

Figura 8 Scratch 3.0 dispone di estensioni, cioè collezioni extra di blocchi che aggiungono nuove funzionalità a quelle delle categorie base. Un'estensione consente anche di programmare la scheda micro:bit di cui si occupa il secondo modulo del libro

Sprite e script

Uno **sprite** è rappresentato da un'immagine che si può muovere su uno sfondo (background) ed è caratterizzato da:

- un **nome**:
- una coppia di **coordinate** (*x*, *y*) che rappresentano la posizione sullo stage del "centro" del suo costume corrente:
- una direzione in cui muoversi;
- una raccolta di **script**:
- un guardaroba di **costumi**, che possono essere indossati solo uno per volta;
- una collezione di **suoni**;
- eventuali **effetti grafici** che modificano la forma e l'apparenza del costume che lo sprite indossa sullo stage.

Uno script è composto da uno o più blocchi impilati in ordine l'uno sull'altro e rappresenta del codice, associato a uno sprite o allo stage, che può essere eseguito a sé stante.

Per eseguire un singolo blocco o uno script è sufficiente fare clic su di esso. Facendo clic sulla bandierina sopra lo stage avviamo l'esecuzione di tutti gli script che iniziano col blocco **quando si clicca su bandierina verde** (come nel **Progetto 1**).

Un programma (o progetto) Scratch combina assieme lo stage e gli sprite con le loro proprietà (immagini, suoni e le caratteristiche sopra elencate) e le loro procedure, cioè gli script che ne definiscono il comportamento.

Categorie dei blocchi

Ogni **istruzione** è costituita da un **blocco colorato**. Alle **9 categorie base** è possibile aggiungere alcune **estensioni**, che presentano altri blocchi con funzionalità specifiche. Per esempio, facendo clic sul pulsante **Aggiungi un'Estensione** (vedi Figura 5, punto 16) e poi sulle estensioni **Musica** e **Penna** (Figura 8), aggiungiamo all'ambiente dell'editor alcuni blocchi che useremo nei prossimi progetti.



Nella tabella che segue, oltre alle categorie base, vengono descritte le due estensioni **Penna** e **Musica** e per ciascuna viene fornito un esempio di blocco specifico. Nel corso del modulo approfondiremo le loro funzionalità.

CATEGORIA	DESCRIZIONE	ESEMPIO DI BLOCCO
Movimento	Blocchi che si occupano del movimento e del posizionamento degli sprite, basandosi su direzione e coordinate x e y. Lo stage non si muove, quindi non presenta questa categoria!	fai 10 passi
Aspetto	Blocchi per creare fumetti e definire come appare lo stage e se e come appaiono gli sprite. Ci sono istruzioni per i fumetti, per il cambio dei costumi e degli sfondi e per l'applicazione di effetti grafici .	dire Ciaol per 2 secondi
Suono	Blocchi per i suoni con cui riprodurre file audio nei formati MP3 e WAV, gestire il volume , applicare effetti sonori ecc.	avvia riproduzione suono. Meow 💌
Situazioni	Blocchi che descrivono situazioni che si attivano al verificarsi di un evento come, per esempio, l'avvio dell'esecuzione al clic sulla bandierina verde. Molti blocchi di questo tipo sono "a cappello", cioè costituiscono sempre il blocco iniziale di uno script .	manthir at the Lorent 🎁
Controllo	Blocchi che controllano il flusso esecutivo degli script attraverso pause , ripetizioni di gruppi di blocchi, blocchi condizionali ecc.	HEER 10 1116L
Sensori	Blocchi per l' input dall'esterno (tastiera e mouse) e di interazione interna tra elementi. È possibile, per esempio, associare un blocco di controllo a un sensore di colore per rilevare se uno sprite tocca un certo colore.	sta toccando il colore
Operatori	Blocchi per utilizzare nei progetti funzioni matematiche e booleane e funzioni per lavorare con le stringhe (sequenze di caratteri).	
Variabili	Blocchi per memorizzare i dati del programma (per esempio, il punteggio in un gioco) in variabili (valori singoli) e liste (elenchi di valori).	porta la mia variabilia 🕶 a 0
I Miei Blocchi	Si possono creare blocchi personalizzati , o procedure, cioè script che possono essere richiamati e riutilizzati quando occorre.	definisci il mio nuovo biocco
Penna	Blocchi associati a una penna (pen) che permette di disegnare in più modalità sullo stage, usando come riferimento il "centro" del costume corrente dello sprite.	porta colore penna a
J.I Musica	Blocchi per suonare con alcuni strumenti musicali e gestire il ritmo della musica. È possibile suonare note musicali , specificando altezza e durata.	passa a strumento (1) Plano 🕶



«There are many extensions of Scratch and some alternative projects like Snap!, an extended reimplementation of Scratch (snap.berkeley.edu) that runs in your browser.»

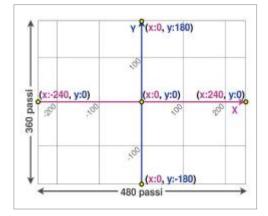


Figura 9 Il sistema di coordinate dello stage di Scratch

Il sistema di riferimento di Scratch

Nel gioco della **battaglia navale**, per posizionare le proprie navi e colpire quelle dell'avversario, si usa un sistema di riferimento relativo alle celle (per esempio, si può "sparare un colpo" nella cella B-5).

Scratch usa un sistema di riferimento per lo stage formato da una griglia di celle **larga 480 passi e alta 360 passi** (*Figura 9*). Il passo è l'unità di misura delle distanze e in modalità di visualizzazione dello stage normale (*Figura 5, punto 14*) corrisponde a un pixel (contrazione di *picture element*, o elemento di immagine del display).



La **coordinata x** indica la distanza in orizzontale del **centro del costume** dal **centro dello stage** ed è data da un numero positivo a destra e negativo a sinistra. La **coordinata y** indica la distanza in verticale ed è data da un numero positivo in alto e un numero negativo in basso. Un nuovo sprite parte dal centro dello stage. Quando spostiamo uno sprite sullo stage, Scratch aggiorna automaticamente le sue coordinate.

Ogni sprite punta in una **direzione** che indica dove va quando si sposta in avanti. Scratch usa un **sistema angolare orario**. Uno sprite nuovo punta verso destra (come indicato dalla freccia bianca in *Figura 10*).

Sotto lo stage c'è il pannello comandi, che, oltre al nome e ai pulsanti per apparire o non apparire sullo stage, presenta le caratteristiche spaziali appena discusse dello sprite selezionato (Figura 11).



0° su

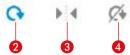


Figura 10 Il sistema per le direzioni di Scratch

- 1. Direzione corrente dello sprite
- 2. Il costume dello sprite può ruotare
- 3. Il costume può solo voltarsi a destra e a sinistra
- 4. Il costume resta fisso (non ruota né si volta)

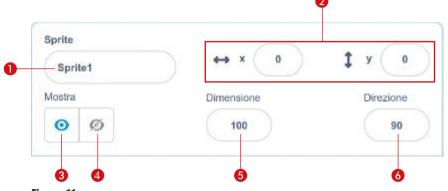


Figura 11
Pannello comandi con le proprietà dello sprite selezionato (Sprite1)

- 1. Nome dello sprite
- 2. Coordinate dello sprite
- 3. Visualizza
- 4. Nascondi

- 5. Dimensione (in percentuale rispetto alla dimensione originale del costume)
- 6. Direzione in cui punta lo sprite (cliccando si apre il pannello di *Figura 10*).

PROGETTO 2

IL VIAGGIO DEL VELIERO

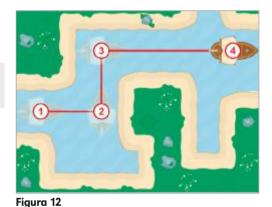


Realizza un'animazione della navigazione del veliero lungo il percorso di *Figura 12*.

Per realizzare il programma è necessario costruire la scena e progettare la sequenza degli spostamenti e dei cambi di direzione. Inoltre, per apprezzare lo spostamento, è opportuno inserire delle **pause** tra un'azione e l'altra. Senza pause, infatti, il computer eseguirebbe tutte le azioni troppo velocemente per riuscire a osservarle.

Nei contenuti digitali del volume (hoepliscuola.it), nella cartella A1-II viaggio del veliero, trovi le immagini dello sfondo e del veliero.

- 1 Crea un nuovo progetto con il nome A1-Il viaggio del veliero. Se sei già nell'editor, seleziona Nuovo dal menu File.
- 2 Cancella lo sprite del gatto facendo clic su □ nella sua miniatura, oppure facendo clic destro sulla miniatura e scegliendo Cancella dal menu a comparsa (Figura 13).
- **3** Sposta il mouse su **Scegli uno Sprite** (Figura 5, punto 24) e fai clic sull'icona superiore **Importa Sprite** (Figura 14) per creare un nuovo sprite a partire da un'immagine. Seleziona **boat.png** nella cartella **A1-II viaggio del veliero**.



Passaggi per far muovere il veliero lungo il percorso che va da 1 a 4



Figura 13
Facendo clic destro sulla miniatura di uno sprite si può duplicarlo, cancellarlo o esportarlo, salvando sul computer un file con estensione .sprite3 che si può importare e riutilizzare in altri progetti



Figura 14 Comandi per la creazione di un nuovo sprite

- 4 In maniera analoga, sposta il mouse su Scegli uno Sfondo (vedi Figura 5, punto 26) e fai clic sull'icona superiore Importa Sfondo per caricare l'immagine sea.png dalla cartella A1-Il viaggio del veliero.
- 5 Seleziona lo sprite, seleziona la categoria e trascina nell'area degli script il blocco iniziale con la bandierina verde.
- 6 Sposta la nave in corrispondenza della tappa n. 1 del percorso (*Figura 12*). Le coordinate non devono necessariamente coincidere con quelle di *Figura 15*. Seleziona la categoria e collega il blocco vai a al blocco precedente.
- 7 Seleziona la categoria e aggiungi il blocco attendi 1 secondi (la traduzione corretta sarebbe "1 secondo").
- 8 Aggiungi un blocco di movimento fai 10 passi e sostituisci il valore 10 con 120.
- 9 Aggiungi un blocco di attesa di un 1 secondo e un blocco ruota in senso antiorario di 15 gradi e sostituisci il valore 15 con 90, in modo che la nave raggiunga la tappa n. 2 del percorso e punti verso l'alto.
- **10** Completa il programma per far compiere al veliero le tratte rimanenti (*Figura 15*).
- 11 Salva il programma. Fai clic sul pulsante 🔀 ed esegui il programma a schermo intero.

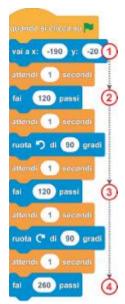


Figura 15Script per far
navigare il veliero



La struttura di controllo sequenza

Nei programmi precedenti abbiamo utilizzato la **struttura di controllo sequen- za**. Per capire meglio di cosa si tratta, è innanzitutto necessario premettere la definizione di costrutto sintattico.

Un costrutto sintattico, o grammaticale, indica come disporre gli elementi di un linguaggio (parole, blocchi ecc.) per formare frasi (nel nostro caso script o parti di script) corrette da un punto di vista di costruzione.

Non è detto tuttavia che una frase sintatticamente corretta sia anche valida a livello di significato.

Per esempio, la frase "Il cane calcia il pensiero" è corretta come costruzione, ma non ha molto senso.

Una struttura di controllo è un costrutto sintattico che controlla il flusso esecutivo, stabilendo l'ordine di valutazione ed esecuzione dei passi di un procedimento o delle istruzioni di un programma.

La struttura di controllo sequenza stabilisce l'esecuzione di tutte le istruzioni in modo ordinato, una dopo l'altra.

Con riferimento al **Progetto 2**, cambiando l'ordine dei blocchi dello script del veliero cambia lo svolgimento dell'azione, anche se il risultato finale potrebbe non cambiare.

Per esempio, facendo muovere la nave prima in alto e poi a destra, l'animazione cambierebbe (il veliero, tra l'altro, si muoverebbe anche sulla terraferma), ma al termine la nave si ritroverebbe comunque in posizione 4 di *Figura 12*.



Sul sito di Scratch si possono condividere i propri progetti e scaricare i progetti condivisi dagli altri utenti. Progetti che possiamo eseguire, modificare e remixare, o semplicemente studiare per capire come sono stati fatti. Infine, se si è registrati, si possono creare gallerie di progetti e usare alcune classiche funzionalità delle applicazioni social come favourite, like e follower.

PROGETTO 3

ORSO CHE CAMMINA

Crea il programma A1-Orso che cammina.sb3

Realizza un programma per far camminare un simpatico orso bruno avanti e indietro all'esterno della sua grotta (*Figura 16*), usando la galleria di immagini che Scratch mette a disposizione per i costumi degli sprite e dei fondali (*backdrop*) dello stage.

- 1 Crea un nuovo progetto con il nome A1-Orso che cammina e cancella lo sprite del gatto, facendo clic sulla □ nella sua miniatura (d'ora in avanti non lo specificheremo più).
- 2 Fai clic su Scegli uno Sfondo e seleziona dalla galleria lo sfondo Mountain (Figura 17). Per trovare velocemente l'immagine puoi usare la funzione di ricerca o fare clic sull'etichetta Esterni.
- **3** Fai clic su **Scegli uno Sprite** per aprire la galleria degli sprite di Scratch. Sposta il mouse su **Bear-walking** per attivare l'animazione dell'orso che cammina e fai clic sull'immagine per aggiungerlo al progetto.

L'orso porta con sé un **guardaroba di 8 costumi** che rappresentano la sua camminata. Il nostro compito è quello di far muovere l'orso sullo stage e contemporaneamente riprodurre in sequenza i costumi, intervallandoli con una breve attesa per simulare i **fotogrammi** di un **cartone**



Figura 16 L'orso bruno cammina avanti e indietro davanti alla tana



Figura 17
Galleria degli sfondi (nella barra superiore si trovano la casella di ricerca e un elenco di categorie per filtrare le immagini)

animato. Per farlo, useremo un blocco della categoria Aspetto che cambia l'abito di scena di uno sprite. Selezionando la scheda (vedi Figura 5, punto 12 a pag. 4), possiamo vedere quali costumi sono presenti nel guardaroba dello sprite e il loro ordine. Ogni costume ha un nome e un numero (in questo caso i costumi sono numerati da 1 a 8).

- 4 Dalla categoria Controllo, trascina nell'area degli script il blocco iniziale.
- **5** Seleziona la categoria **Movimento** e aggiungi allo script il blocco usa stille retazione simistra-destra. In questo modo, quando lo sprite cambia direzione, il costume non ruota ma **si capovolge solo in orizzontale**, rivolgendosi a destra o a sinistra. Puoi ottenere lo stesso effetto agendo su **Direzione** nel pannello di controllo dello sprite (*vedi Figura 10, punto 3* a pag. 8).
- 6 Seleziona la categoria Controllo e aggiungi il blocco per sempre. Si tratta di un blocco a C che può includere altri blocchi, che risultano dipendenti da esso. In questo caso, i blocchi contenuti vengono eseguiti fino a quando non si ferma il programma, per esempio facendo clic sul pulsante Ferma tutto.



Figura 18Script del progetto
"Orso che cammina"



- 7 Seleziona la categoria Movimento e inserisci nel blocco precedente i blocchi fai 10 passi e rimbalza quando tocchi il bordo. Con questi blocchi l'orso avanza di 10 passi nella direzione in cui sta puntando, voltandosi dalla parte opposta quando arriva a toccare il bordo destro o sinistro dello stage.
- 8 Seleziona la categoria e aggiungi il blocco passa al costume seguente. Quindi, dopo lo spostamento, l'orso indossa il costume successivo nel guardaroba e dopo l'ultimo riparte dal primo.
- **9** Completa lo script (*Figura 18*), inserendo il blocco **attendi 1 secondi** e digitando **0.2** per far aspettare lo sprite 2 decimi di secondo prima di ripetere le azioni all'interno del ciclo.
- 10 Salva ed esegui il programma.
- **11** Mentre il programma è in esecuzione, prova a modificare la velocità dell'animazione, cambiando il numero dei passi e il tempo di attesa. L'effetto è immediato e non richiede di far ripartire il programma.



Figura 19 I tre blocchi per costruire una ripetizione con Scratch

La struttura di controllo ripetizione

Nell'ultimo programma, oltre alla **sequenza**, abbiamo utilizzato la **ripetizione**.

→ La struttura di controllo ripetizione (o iterazione, o ciclo) ripete una sequenza di blocchi per sempre, un certo numero di volte, oppure fino a quando una certa condizione non diventa vera.

Per realizzare questa struttura di controllo, Scratch mette a disposizione **tre tipi di blocco a C**, che ripetono l'esecuzione dei blocchi contenuti (*Figura 19*). Questi ultimi costituiscono il "**corpo**" **della ripetizione**.

Concludiamo l'unità con un programma con più sprite e con l'aggiunta di un tocco di **casualità** nell'animazione.

PROGETTO 4

IN FONDO AL MAR



Realizza un programma per far nuotare alcuni pesci avanti e indietro in un fondale marino. All'inizio, i pesci appaiono in posizioni casuali e poi si muovono con velocità variabile (Figura 20).

- 1 Crea un nuovo progetto con il nome A1-In fondo al mar.
- 2 Fai clic su Scegli uno Sfondo e seleziona dalla galleria lo sfondo Underwater 1.
- 3 Fai clic su Scegli uno Sprite per aprire la galleria degli sprite di Scratch e aggiungi lo sprite Fish, che ha 4 costumi di pesci diversi (il primo costume fish-a è quello di un pesce pagliaccio).
- 4 Trascina nell'area degli script il blocco iniziale.
- 5 Seleziona la categoria Movimento e aggiungi il blocco usa stile rotazione sinistra-destra.
- 6 Aggiungi il blocco di Movimento
- **7** Seleziona la categoria **Controllo** e aggiungi il blocco **per sempre**.
- 8 Seleziona la categoria Movimento e inserisci nel ciclo i blocchi fai 10 passi e rimbalza quando tocchi il bordo.
- 9 Seleziona la categoria personi, trascina nella casella del blocco fai 10 passi il blocco numero a caso tra 1 e 10 e nella seconda casella digita 6. In questo modo, lo sprite avanzerà con spostamenti brevi variabili tra 1 e 6 passi (Figura 21).
- 10 Ripeti il passo 3 che aggiunge il secondo sprite Fish2.
- **11** Seleziona la scheda **Costumi** di **Fish2** e fai clic sul secondo costume (**fish-b**).
- **12** Fai clic sulla miniatura di **Fish** e poi attiva la scheda **Codice**. Copia lo script trascinandolo dal primo blocco sulla miniatura di **Fish2** (*Figura 22*).
- **13** Completa il programma, ripetendo i **passi** dal **10** al **12** per creare gli altri due sprite dei pesci.
- 14 Salva ed esegui il programma.



Figura 20 Stage del programma "In fondo al mar"



Figura 21 Lo script di Fish è presente anche negli altri pesci di "In fondo al mar"



Fig Tras sull

Figura 22 Trascinando uno script da Fish sulla miniatura di Fish2 ne creiamo una copia in quest'ultimo

PALESTRA DELLE COMPETENZE



A domanda... rispondo!

VERO FALSO



- 1 Scratch non è un linguaggio di programmazione a blocchi.
- 2 Il progetto Scratch è composto da un linguaggio, un editor, una piattaforma cloud e una community.
- 3 Scratch prevede solo una modalità di lavoro online.
- 4 Scratch non permette di creare gli sprite senza alcun costume.
- 5 Uno script è un semplice insieme di blocchi.

9 Lo stage ha solo alcuni blocchi di movimento.

8 Uno sprite nuovo punta verso destra.

- 6 La struttura di controllo sequenza consente di ripetere uno o più blocchi.
- 7 Due blocchi per implementare la ripetizione in Scratch sono:





VF

F

F

F

- VF
- V



Progetto e creo!

Utilizza Scratch per creare i programmi che risolvano i seguenti problemi.

1 Crea il file Nella savana.sb3

Scrivi un programma che faccia camminare avanti e indietro una zebra e una giraffa nella savana (*Figura 23*). Le immagini per lo sfondo e gli sprite sono presenti nella libreria di Scratch.

2 Crea il file Posizionamento casuale.sb3

Scrivi un programma che faccia apparire per mezzo secondo lo sprite **Ball**, disponibile nella libreria di Scratch (*Figura 24*), in un punto casuale dello stage.

3 Inventa e crea il file <Nome del programma>.sb3

Inventa, progetta e realizza un programma, con un nome a tua scelta, che realizzi un'animazione sul modello del **Progetto 2**.



Figura 23 Stage del programma "Nella savana"



Figura 24 Miniatura dello sprite Ball