

## 3. CODICE = CODING

L'arte di programmare è sempre stata appannaggio di ingegneri esperti, ma oggi, grazie all'evoluzione dei computer e dei linguaggi di programmazione, è possibile sperimentarla anche a scuola. Sono stati creati molti tipi di linguaggi ma quelli che si sono maggiormente affermati sono quelli detti **a blocchi**, chiamati così perché costituiti da una serie di forme a incastro, un po' come i pezzi di un puzzle, che combinandosi tra loro realizzano una vera e propria programmazione. Ciò è possibile perché ogni blocco, in realtà, è un contenitore riempito con una serie di informazioni che rappresentano il **codice** di programmazione. L'azione di comporre questi tasselli si chiama **coding**, ma in pratica non dobbiamo scrivere nulla, semplicemente incastrare questi pezzi di codice per ottenere l'azione desiderata. Questo è fondamentale per uno studente alle prime armi perché elimina la difficoltà della sintassi e fornisce una risposta grafica immediata.

### ■ L'apprendimento ludico di Scratch

Il programma più diffuso per sperimentare il coding nelle scuole prende il nome di **Scratch** ed è nato a Boston presso il Massachusetts Institute of Technology (MIT) nel 2006 grazie all'intuizione di Mitchel Resnick. L'idea nacque con l'intento di creare uno strumento che permettesse di imparare a programmare in maniera semplice e divertente fin dalla più tenera età. Scratch è, infatti, accessibile a tutti, facilissimo da utilizzare e, cosa molto importante, è gratuito, open source, multiplatforma e disponibile sia in versione online che offline. Dispone, inoltre, di una enorme community nella quale poter trovare risorse, progetti, programmi di qualunque genere da utilizzare liberamente senza dover richiedere costose licenze o autorizzazioni. Con Scratch è possibile in maniera molto intuitiva realizzare animazioni, cartoni animati, puzzle, quiz ma anche piccoli giochi arcade.

È un modo semplice per avvicinare i giovani al mondo della programmazione e per *insegnare a pensare in maniera creativa utilizzando un approccio ludico*.

### ■ Installiamo Scratch

Abbiamo detto che è possibile utilizzare Scratch *online*, accedendo alla pagina <https://scratch.mit.edu> e registrandosi alla community per poter condividere contenuti e risorse, oppure in modalità *offline* scaricando

l'app disponibile per Windows, macOS, ChromeOS, iOS e Android sul proprio dispositivo partendo dalla pagina <https://scratch.mit.edu/download>. Una volta scaricato il software, basta seguire le semplici istruzioni, differenti per piattaforma, per installarlo, dopodiché sarai pronto a utilizzarlo.

Per poter utilizzare in maniera fattiva il programma è necessario capire quali sono gli elementi che questo ci fornisce e come sono organizzati sullo schermo. Dovremmo imparare a conoscere quella che in gergo tecnico viene chiamata *interfaccia grafica*.

Lo schermo è così diviso: la **Barra degli Strumenti o Toolbar**, i **Tabs (codice, costumi, suoni)**, l'**Area degli script**, il **Pannello Stage**, il **Pannello Informazioni**, il **Pannello Sfondi** e il **Pannello con la lista degli Sprite del progetto**.

## GLI STAGE E GLI SPRITE

Prima di iniziare è importante fare conoscenza con alcuni dei termini tipici di Scratch, come ad esempio gli Stage e gli Sprite. Sembrano termini complessi ma in realtà rappresentano gli oggetti che si usano sul programma.

Gli **Stage** sono il palcoscenico, ossia l'ambientazione nella quale si svolge la nostra scena; mentre gli **Sprite** sono gli elementi grafici (personaggi o oggetti) su cui agisce il codice del programma.

## LA BARRA DEGLI STRUMENTI (TOOLBAR)

La barra degli strumenti posta in alto nell'interfaccia grafica del programma, accanto al logo di Scratch, riporta alcuni menu a discesa e pulsanti:

- una piccola **icona a forma di mondo** attraverso la quale è possibile selezionare la lingua per il programma;
- il menu **File** consente di accedere a tre differenti funzionalità:
  - **Nuovo** – per creare un nuovo progetto;
  - **Carica dal computer** – per aprire un progetto già salvato sul proprio computer;
  - **Salva sul tuo computer** – per salvare sul proprio pc un progetto sul quale si sta lavorando.
- il menu **Modifica** consente di accedere a due funzionalità:
  - **Ripristina** – che permette di annullare l'ultima azione compiuta;
  - **Attiva o disattiva modalità turbo**, ossia accelera e rende più colorate le sequenze create con molti script.

- il menu **Tutorial**, quello con la piccola lampadina accesa, permette di accedere a moltissimi progetti condivisi che spiegano tutti i comandi del programma e gli elementi dell'interfaccia;
- il **Campo di testo** che contiene il nome del nostro progetto che possiamo modificare in qualunque momento;
- il pulsante **Condividi** per permetterci di condividere il nostro progetto con altri membri della community;
- Infine il pulsante **Vai alla Pagina del Progetto** che ci consente di accedere ai nostri progetti salvati.

## I TABS

Immediatamente sotto la barra del menu troviamo tre differenti **Tabs**, ossia delle linguette (dette di navigazione) che permettono all'utente di spostarsi tra un gruppo di controllo e un altro. Hanno forma rettangolare e presentano all'interno un nome che descrive il loro contenuto. Cliccandovi sopra verranno resi visibili tutti gli elementi contenuti al loro interno. I Tab sono Codice, Costumi e Suoni ma nel caso sia stato selezionato uno Stage e non uno Sprite, il Tab Costumi diventa Sfondi.

## IL TAB CODICE

Il primo Tab è quello denominato **Codice** che si divide a sua volta in tre colonne (vedi esempio qui a destra):

- la **prima colonna**, evidenziata in rosso, contiene una serie di pallini colorati i quali dividono i comandi del programma in **blocchi** in base al loro differente scopo:
  1. **movimento**, ossia tutte le istruzioni per far muovere gli Sprite sulla scena;
  2. **aspetto**, ossia tutte le istruzioni per far parlare e pensare gli Sprite con la tecnica del fumetto e di cambiare costume;
  3. **suono**, ossia tutte le istruzioni per inserire suoni, effetti o parole sulla scena;
  4. **situazioni**, ossia tutte le istruzioni per avviare uno script, inviare messaggi ad altri script, ecc.;
  5. **controllo**, ossia tutte le istruzioni per consentire agli Sprite di ripetere azioni, attendere specifici eventi o comandi, ecc.;
  6. **sensori**, ossia tutte le istruzioni per ricevere input dall'esterno, come ad esempio clic del mouse o la pressione dei tasti sulla tastiera, oppure dall'interno per eseguire un programma per

consentire ad uno Sprite di reagire in maniera specifica a un singolo evento;

7. **operatori**, ossia tutte le istruzioni per eseguire operazioni matematiche, manipolare testi, valutare determinate condizioni...;
  8. **variabili**, ossia tutte le istruzioni residenti nella memoria del computer che rendono il programma capace di manipolare i dati (parole, caratteri, numeri...);
  9. **i miei blocchi**, ossia tutte le istruzioni che permettono al programmatore di realizzare nuovi mattoncini con nuove istruzioni.
- la **seconda colonna** è quella evidenziata in giallo; contiene i blocchi o i mattoncini relativi alla categoria selezionata, evidenziati anch'essi con lo stesso colore del pallino della colonna precedente. Ciascun elemento di questa colonna è una **sequenza di comandi** che possono essere utilizzati semplicemente trascinando il mattoncino prescelto nella successiva colonna, quella evidenziata in verde detta degli script o del Codice.
  - la **terza colonna** è quella evidenziata in verde e rappresenta l'**area degli Script o del Codice**; è la vera e propria **area di programmazione** dove mattoncini della seconda colonna vengono disposti in sequenza realizzando lo script del nostro programma.

## I TABS COSTUMI E SFONDO

Il secondo Tab è quello denominato **Costumi** e ci consente di accedere alla scheda con la quale potremo modificare l'aspetto del nostro Sprite scegliendo tra quelli contenuti nella libreria di Scratch, caricando l'immagine dal nostro computer o da internet, acquisendo l'immagine con la videocamera oppure disegnandone noi uno integralmente.

Questa scheda ci consente di agire praticamente su tutti gli aspetti dello Sprite modificandone ogni elemento, come ad esempio il colore, la forma, il bordo e così via.

Se abbiamo selezionato uno Stage (fondi) il Tab diventa **Sfondi** e il funzionamento è praticamente lo stesso del Tab Costumi, consentendo la modifica di ogni aspetto dello sfondo prescelto.

## IL TAB SUONI

Il terzo Tab è quello denominato **Suoni** e, come dice il suo nome, ci consente di aggiungere alla nostra azione suoni che possiamo recuperare dalla libreria di Scratch, caricandoli dal nostro computer o dalla rete, registrandone di nuovi attraverso un microfono.

Il pannello consente, inoltre, di modificare moltissimi aspetti del suono che abbiamo inserito come la velocità, il volume, le dissolvenze e molto altro.

## LA FINESTRA STAGE E I PANNELLI SPRITE LIST E STAGE LIST

La destra dell'interfaccia grafica è occupata da tre pannelli che danno informazioni sugli oggetti in uso sulla scena.

Il primo evidenziato in rosso è la **finestra Stage**, dove si svolge la scena e dove possiamo vedere scorrere i personaggi e gli effetti della programmazione sullo sfondo prescelto. Dobbiamo ricordarci che il nostro palcoscenico (**Stage**) ha una dimensione ben precisa che dovremo rispettare nello svolgimento della nostra animazione. La dimensione dello Stage è di 480 passi in larghezza e di 360 passi in altezza. La misurazione viene effettuata in passi perché l'obiettivo è quello di animare lo Sprite, ossia il personaggio, l'animale o l'oggetto prescelto. Il secondo è il **pannello Sprite List** evidenziato in giallo dove possiamo trovare le informazioni sui personaggi scelti e da dove possiamo cambiare la loro posizione, dimensioni, direzione di spostamento e renderli visibili o invisibili.

Infine, all'estrema destra troviamo il **pannello Stage List**, evidenziato in verde, dove possiamo vedere lo sfondo selezionato, importarne uno nuovo o permettere al programma di inserirne uno a sorpresa.

## CIAK SI GIRA

Proviamo subito a interagire con Scratch per capire come è facile realizzare uno script per creare un'animazione, un gioco o qualunque altra cosa abbiamo in mente.

Scratch utilizza la tecnica del *Drag and Drop* ossia del prendere, trascinare e rilasciare gli oggetti. Si tratta di selezionare i mattoncini colorati presenti nel Tab Codice e trascinarli nell'Area degli Script. Questi si incastreranno l'uno dietro l'altro come le tessere di un puzzle riproducendo la sequenza di un algoritmo.

## MUOVIAMO I PRIMI PASSI

*Elementi della scena*

Sprite: **1**

Sfondo: **nessuno**

Categoria di codici utilizzati: **Movimento, Suono, Situazioni**

Per prima cosa **creiamo un nuovo file** di lavoro; apparirà la schermata Stage bianca con il gatto di Scratch al centro.

1. Trasciniamo dal **Tab Codice**, categoria **Movimento** il primo mattoncino, quello con il numero dei passi, nell'**Area degli Script**. Decidi quanti passi il nostro personaggio dovrà fare (abbiamo inserito 40 passi). Proviamo adesso a inserire un suono per sottolineare i passi del nostro gatto. Dal **Tab Suoni**, utilizzando il pulsante in basso a sinistra con l'altoparlante con il simbolo +, carichiamo dalla libreria di Scratch il suono "Boing".
2. Trasciniamo dal **Tab Codice**, categoria **Suono** il secondo mattoncino, quello per riprodurre un suono e selezioniamo dall'elenco *Boing*, quello che abbiamo appena scaricato.
3. Duplichiamo i due mattoncini che abbiamo inserito nell'Area degli Script e incastriamoli in sequenza ai primi due. Adesso, se clicchiamo su qualunque mattoncino, il nostro gatto inizierà a muoversi riproducendo un suono ad ogni passo.
4. Trasciniamo dal **Tab Codice**, categoria **Situazioni** il primo mattoncino, quello con la bandierina verde e posizioniamolo in cima alla pila di mattoncini che abbiamo già realizzato. In questo modo, l'animazione inizierà solo quando cliccheremo con il mouse la bandierina verde sopra l'area Stage.

## ATTIVITÀ GUIDATA · Il volo del pipistrello

*Elementi della scena*

Sprite: **1**

Sfondo: **1**

Categoria di codici utilizzati:

**Movimento, Situazioni, Penne**

Come abbiamo visto, è possibile far spostare all'interno dello Stage qualunque Sprite liberamente, l'importante è non superare i limiti dell'area disponibile. Seguiamo adesso il libero volo di un pipistrello a zig-zag nel buio della città metropolitana.

Per prima cosa creiamo un nuovo file di lavoro. Apparirà la schermata Stage bianca con il gatto di Scratch al centro.

Dal pannello **Stage List**, scegliamo un nuovo sfondo, in questo caso una visione notturna di una città metropolitana (Night City) e dalla **Sprite List** il nostro protagonista (Bat).

Stabiliamo il percorso che Bat dovrà fare zigzagando tra gli edifici della nostra assonnata città. Inizialmente il pipistrello si troverà nella posizione ( $x=-150$ ;  $y=100$ ) fissata con il comando **Vai** dalla categoria **Movimento**, ma con rapidi battiti di ali si sposta passando per ( $x=-45$ ;  $y=-110$ ), ( $x=50$ ;  $y=0$ ), ( $x=140$ ;  $y=-130$ ) ed infine ( $x=170$ ;  $y=110$ ) grazie

al comando **Scivola** sempre da **Movimento**. In questo caso dovrai anche fissare il tempo in cui Bat percorrerà le distanze tra i punti (2 secondi).

È possibile anche visualizzare il percorso compiuto dal nostro pipistrello utilizzando i blocchi **Penna** che trovate in **Aggiungi un'estensione** in basso a sinistra nella schermata di Scratch:

Il blocco "pulisci" cancella tutte le righe che sono state tracciate dai blocchi Penna e quello "penna giù" posa la penna sull'area da lavoro per disegnare.

## INSERIRE ISTRUZIONI E VEDERE LE RISPOSTE (INPUT/OUTPUT)

Per far interagire il programmatore (noi) con la macchina servono istruzioni precise dette di **input** (inserimento di dati, risposte a domande, ecc.) e di **output** (scritte visualizzate sullo schermo, suoni, ecc.).

Scratch gestisce le istruzioni di input/output attraverso dei blocchi particolari.

### INPUT

La logica dell'operazione di input da parte di Scratch è la seguente. Il programma:

1. **chiede** all'utente di inserire l'informazione;
2. **attende** finché l'utente non inserisce l'informazione che sarà salvata in un blocco chiamato "risposta".

Una volta utilizzato il blocco "chiedi e attendi" ed eseguito il programma, si aprirà una barra nella parte inferiore dello Stage in cui potrai inserire l'informazione richiesta.

Una volta digitata, conferma, premendo il tasto INVIO sulla tastiera, nella vignetta sopra Scratch comparirà per il tempo stabilito il nome che hai digitato.

### OUTPUT (USCITA)

La maggior parte dei blocchi di output si trovano nelle categorie **Aspetto** e **Suono**. Tra i primi troviamo "**dire...**" e "**pensa...**" la cui differenza dipende solo dal tipo di vignetta che sarà mostrata mentre tra i secondi troviamo quelli relativi alla riproduzione di suoni:

## ATTIVITÀ GUIDATA · Calcola l'area e il perimetro di un quadrato

*Elementi della scena*

Sprite: **1**

Sfondo: **1**

Categoria di codici utilizzati: **Situazioni, Aspetto, Operatori, Sensori**

Metti assieme le conoscenze apprese fino ad ora.

Utilizza i blocchi di input/output per creare un programma che calcoli l'area e il perimetro di un qualsiasi quadrato a cui aggiungeremo alcuni **Operatori**.

Dopo aver chiesto di inserire la lunghezza del lato del quadrato, utilizza il blocco "**moltiplicazione**" della categoria **Operatori** per calcolare l'area del quadrato (lato per lato).

Il blocco "risposta" contiene la lunghezza del lato, quindi basterà moltiplicarla per sé stessa. Infine utilizza il blocco di output testuale "**dire**" per visualizzare il risultato.

Successivamente calcola il perimetro, utilizzando un altro blocco "**moltiplicazione**" e un altro blocco "**dire**".

## ATTIVITÀ GUIDATA · Calcola la lunghezza della circonferenza e l'area di un cerchio con raggio variabile

*Elementi della scena*

Sprite: **1**

Sfondo: **1**

Categoria di codici utilizzati: **Situazioni, Aspetto, Operatori, Sensori**

Anche in questo caso utilizziamo i blocchi di input/output. Scratch ci chiederà la lunghezza del raggio attraverso il blocco **chiedi** della categoria **Sensori**. Attraverso gli **Operatori**, risolvi il problema e tramite i blocchi di output (**dire**) della categoria **Aspetto** visualizzalo sullo schermo.

(N.B.: il valore di  $\pi$  è stato approssimato a 3,14)

### GLI OPERATORI

I blocchi **Operatori** possono essere concatenati per creare operazioni complesse.

Negli esercizi precedenti abbiamo osservato un loro utilizzo come ad esempio con più blocchi "moltiplicazione" per risolvere il problema del calcolo della lunghezza della circonferenza e del calcolo dell'area del cerchio.

Trascinando, quindi, un secondo blocco "moltiplicazione" all'interno di uno dei due fattori otterrai l'operazione  $[3,14 * 2 * \text{raggio}]$  che risolve

il problema della lunghezza della circonferenza.  
Lo stesso principio vale per il calcolo dell'area del cerchio.

## LE VARIABILI

Le variabili sono dei contenitori dove vengono memorizzati i dati e le informazioni.

Hai già visto e utilizzato una variabile in Scratch (il blocco **"risposta"**) ma a differenza di quelle che vedrai in questo paragrafo, il valore di **"risposta"** non si poteva modificare.

Esistono principalmente due tipi di variabili:

### 1. Le variabili globali

Sono quelle che possono essere scritte e lette in qualsiasi momento e da qualsiasi Sprite del programma. Ad esempio il tempo in un videogioco è una variabile globale, perché ogni volta che perderai una vita dovrai ricominciare e il tempo dovrà essere azzerato. **"Tempo"** è una variabile globale. Anche **"risposta"** è una variabile globale.

### 2. Le variabili locali

Sono variabili che possono essere lette e scritte solo in alcune zone del programma o da determinati Sprite. Sempre nell'ipotesi del videogioco, ogni giocatore (Sprite) potrà perdere o guadagnare solo le proprie vite e non quelle dell'avversario.

**"Vita"** sarà una variabile locale, ovvero al di fuori dello Sprite in cui è stata creata, non esiste!

Fino ad ora potevi effettuare un'operazione alla volta, perché ogni volta che inserivi un'informazione nuova, **"risposta"** veniva sovrascritta. Ora vedrai che grazie alle variabili potrai salvare il contenuto di **"risposta"** e avere sempre a disposizione il dato.

## VARIABILI IN SCRATCH

Per creare una variabile devi spostarti nella categoria **Variabili** e cliccare su **"Crea una Variabile"**. Nella finestra che apparirà digita il nome della variabile e il tipo.

- **Per tutti gli Sprite** (globale);
- **Solo per questo Sprite** (locale).

Conferma cliccando su **"OK"**.

## ATTIVITÀ GUIDATA · Le iniziali

*Elementi della scena*

Sprite: 1

Sfondo: 1

Categoria di codici utilizzati: **Sensori, Variabili, Situazioni, Aspetto, Operatori**

Lo scopo di questo semplice esercizio è ottenere le iniziali del nome e cognome dell'utente.

Crea due variabili che chiamerai "nome" la prima e "cognome" la seconda. Scratch ti chiederà di inserire il tuo nome e lo salverà nella variabile "nome" (porta "nome" a "risposta"). Fai la stessa cosa per il cognome.

Concatena i blocchi "unione" e "lettera 1 di..." per ottenere la risposta desiderata.

## ATTIVITÀ GUIDATA · L'area del triangolo

*Elementi della scena*

Sprite: 2

Sfondo: 1

Categoria di codici utilizzati: **Aspetto, Situazioni, Variabili, Operatori, Sensori, Suono**

In questo esercizio insieme al gattino Scratch metteremo assieme molte delle conoscenze apprese fino ad ora. Utilizzeremo i blocchi di input e output per permettere a Scratch di calcolare l'area del triangolo sulla base dei dati che gli forniremo, le variabili e gli operatori matematici. Per prima cosa dovremo creare dalla categoria **Variabili** tre nuove variabili, che chiameremo "base", "altezza" e "area".

Possiamo anche inserire uno Stage a piacere come sfondo, meglio se a tinta unita.

1. In questo primo blocco di comandi porteremo a zero il valore di tutte e tre le variabili in modo che lo script inizi sempre con i dati azzerati.
2. Nel secondo e terzo blocco impileremo i mattoncini di input "chiedi" con i quali dovremo inserire i dati mancanti, ossia *base* e *altezza*, per far sì che le variabili assumano questi valori. Scratch resterà in attesa fin quando non inseriremo i valori numerici delle due dimensioni.
3. Dopo una breve pausa riflessiva "Pensa..." Scratch mostrerà attraverso gli operatori di calcolo il risultato richiesto accompagnato, se vogliamo, da un suono di applausi per il successo ottenuto.

## STRINGHE

Una stringa in informatica è una sequenza di caratteri con un ordine prestabilito. Con carattere si intende una qualsiasi lettera, numero o simbolo presente sulla tastiera.

Ad esempio, "Ciao", "4321", "Il mio nome è Martina" sono tutte stringhe. Scratch permette di effettuare delle operazioni sulle stringhe, in particolare:

1. l'unione tra due o più stringhe;
2. estrapolare un carattere specifico della stringa conoscendo la sua posizione;
3. sapere da quanti caratteri è costituita la stringa.

## ATTIVITÀ GUIDATA • Scratch ci saluta

*Elementi della scena*

Sprite: **1**

Sfondo: **room 1**

Categoria di codici utilizzati: **Controllo, Aspetto, Sensori, Operatori, Variabili**

Per prima cosa **creiamo un nuovo file** di lavoro e inseriamo Scratch e uno **Stage** per creare la nostra ambientazione.

1. Trasciniamo dal **Tab Codice**, dalla categoria **Situazioni** il primo mattoncino.
2. Dalla categoria **Aspetto** trasciniamo e agganciamo sotto il mattoncino precedente il primo elemento nel quale possiamo inserire il parlato di Scratch, **dire...** dove scriveremo, ad esempio, "*Ciao sono Scratch il gattino di questo programma*" e fissiamo una durata di 2 secondi.
3. Dalla categoria **Sensori** selezioniamo il mattoncino 5 quello che inizia con "**Chiedi**". Qui potremo inserire la domanda che Scratch ci rivolge, ad esempio "*E tu come ti chiami?*". Questo mattoncino farà apparire un campo nel quale potremo scrivere una risposta, in questo caso il nostro nome.
4. Aprendo la categoria **Variabili**, dovremo crearne una nuova attraverso l'apposito pulsante "**Crea una variabile**" che chiameremo "nome". Trascineremo, poi, il primo mattoncino, quello che inizia con "**porta...**" sotto agli altri e sceglieremo tra le variabili disponibili quella che abbiamo appena creato cioè "nome". Riempiremo poi il secondo campo, quello con lo zero con il comando **Risposta** preso dalla categoria **Sensori**. In questo modo, la nostra variabile "nome", verrà associata alla risposta che daremo. In questo come nei precedenti

esempi, è interessante notare come i mattoncini di Scratch siano combinabili tra di loro. Infatti i campi hanno determinate forme che possono essere riempite con altri comandi che hanno quella stessa forma.

5. A questo punto Scratch ci risponde; trasciniamo di nuovo il mattoncino **dire...** dal **Tab Aspetto** e inseriamo nel primo spazio un **Operatore di addizione** (stringa) preso dalla categoria **Operatori**. La risposta potrebbe essere *"Piacere di conoscerti"* a cui assoceremo la variabile *"nome"* il cui valore abbiamo precedentemente associato a quanto abbiamo scritto, cioè al nostro nome. Quindi, Scratch risponderà *"Piacere di conoscerti"* seguito dal nome digitato per il tempo stabilito, in questo caso 2 secondi.
6. Dalla categoria **Controllo**, trasciniamo e agganciamo il mattoncino che inizia con **"ferma"** che ci permetterà di concludere questo script. Se non volete vedere sullo Stage il valore della variabile *"nome"* basterà inserire un nuovo mattoncino dalla categoria **Variabili** quello che inizia con **"Nascondi"** e associare la nostra variabile, ossia *"nome"*.

## I CICLI ITERATIVI

Immagina di dover ripetere un'azione più volte. In base a quello che abbiamo finora appreso basterebbe impilare più mattoncini con lo stesso comando il numero di volte desiderato. Ma questo non è il lavoro ideale perché non ci agevola per nulla.

Per risolvere questo problema, Scratch e i linguaggi di programmazione utilizzano i **cicli iterativi** che consentono di risparmiare righe di codice e velocizzare lo sviluppo del programma.

Si tratta di blocchi che racchiudono al loro interno diverse istruzioni e permettono di ripeterle *n* volte finché non si verifica una condizione di uscita.

I cicli iterativi si trovano all'interno delle variabili di **Controllo** e ne esistono di più tipologie in base al problema da risolvere.

Il ciclo **"per sempre"** ripete infinite volte le istruzioni al suo interno. In genere non si utilizza perché questo ciclo non prevede un'uscita quindi l'operazione viene ripetuta all'infinito.

Il ciclo **"Ripeti n volte"** esegue le istruzioni che si trovano al suo interno *"n* volte" dopodiché prosegue con le istruzioni poste inferiormente. In questo caso la condizione di uscita è il numero di volte in cui vengono ripetute le istruzioni.

Il ciclo **"Ripeti fino a quando"** ti permette di impostare la condizione di uscita che non sia solo numerica. Ad esempio continua a eseguire

le istruzioni al suo interno finché lo Sprite non tocca il bordo o un oggetto oppure fino a che non si verifica un evento (click del mouse, tasto premuto) o finché non si riceve un dato da un sensore.

## ATTIVITÀ GUIDATA · La rana che salta il bicchiere

*Elementi della scena*

Sprite: **2**

Sfondo: **1**

Categoria di codici utilizzati: **Controllo, Suono, Situazioni, Movimento**

Immaginiamo che una rana voglia saltare oltre un bicchiere posto dinnanzi a lei per poi ritornare alla posizione di partenza. Immaginiamo anche che voglia saltare avanti e indietro per 4 volte. Dopo aver scelto lo Stage e i due Sprite (la rana e il bicchiere), stabiliamo la posizione di partenza dei due oggetti. Dopodiché inizieremo a comporre i mattoncini dell'azione. Dopo aver inserito dalla categoria **Situazioni** il mattoncino di partenza che, in questo caso avvierà lo script quando premeremo la barra spaziatrice della tastiera, andremo a posizionare la sequenza di mattoncini di movimento ai quali possiamo aggiungere il gracidio che la nostra rana emetterà durante questi balzi.

1. Indicheremo la direzione di salto "*punta in direzione*" verso destra ossia "90°", la traiettoria verso l'alto che la rana compirà dopo aver spiccato il balzo "*vai a X=-170 e Y=-115*" con indicato il tempo impiegato durante questo salto ossia "1 secondo" e infine la posizione in cui dovrà trovarsi la rana al completamento del salto dopo aver scavalcato il bicchiere.
2. Nel secondo blocco di comandi la rana cambierà direzione di salto "-90°" per poter tornare alla posizione iniziale dopo aver saltato nuovamente e in verso opposto il bicchiere.

Se tu volessi far compiere alla rana 4 volte il salto del bicchiere dovrai duplicare l'intera sequenza di comandi (1 e 2) e impilarli quattro volte sotto il precedente script. Il tutto funzionerebbe perfettamente ma con un numero eccessivo di comandi e un allungamento notevole dello script.

Utilizzando invece un ciclo iterativo dalla categoria **Controllo**, puoi inglobare l'intero script al suo interno e stabilire che questo venga ripetuto un numero n di volte al termine del quale il programma sarà terminato.

## VERO/FALSO E GLI EVENTI CONDIZIONALI

A volte quando eseguiamo un'azione siamo posti di fronte a delle scelte obbligate. Ad esempio posso aprire la porta se ho la chiave altrimenti la porta non potrà essere aperta. La condizione è, quindi, quella di possedere la chiave. Attraverserò l'incrocio se il semaforo è verde altrimenti dovrò fermarmi. La condizione in questo caso è il colore del semaforo. Altra considerazione è che non si possono eseguire contemporaneamente entrambe le istruzioni perché opposte (avrò o non avrò la chiave, il semaforo sarà rosso o verde).

Il costrutto "se" in programmazione è comunemente utilizzato per eseguire blocchi di codice condizionalmente, a seconda che una certa condizione sia vera o falsa. In Scratch il costrutto "se" è rappresentato da un blocco condizionale.

Ecco come puoi utilizzare il costrutto "se" in Scratch:

1. vai al blocco "Controllo" nella barra laterale sinistra di Scratch;
2. trova il blocco "se" e trascinalo nell'area di lavoro;
3. dopo aver posizionato il blocco "se" nell'area di lavoro, avrai bisogno di aggiungere una condizione. La condizione determina se il blocco di codice all'interno del blocco "se" verrà eseguito o meno;
4. vai al blocco "Operatori" nella barra laterale sinistra di Scratch;
5. trova i blocchi che ti permettono di creare una condizione. Ad esempio, puoi utilizzare il blocco "uguale a" per verificare se due valori sono uguali o se uno è maggiore di un altro.

### ATTIVITÀ GUIDATA · Posso entrare nel castello?

*Elementi della scena*

Sprite: 1

Sfondo: 1

Categoria di codici utilizzati: **Controllo, Operatori, Aspetto, Suono, Situazioni, Movimento**

Un cavaliere posto a guardia dell'ingresso del castello controlla chi può entrare o meno e per questo deve verificare l'età dei passanti.

Egli chiederà a ciascuno di noi quanti anni abbiamo e solo nel caso in cui siamo maggiorenni, ossia abbiamo più di 18 anni, il cavaliere ci consentirà di accedere al maniero. La condizione, quindi, per accedere è solo quella di avere più di 18 anni. Lo script inizierà con il cavaliere che ci chiede l'età restando in attesa della nostra risposta (input/output). In base a questa si potranno verificare solo due condizioni.

Nel primo la nostra età sarà maggiore di 18 per cui il cavaliere ci accoglierà all'interno del castello e un suono riprodurrà il cigolio della porta che si apre. Nel secondo, invece, il cavaliere esprimerà il suo diniego a farci entrare chiudendo la porta sbattendola (suono).

### ATTIVITÀ GUIDATA · Confronto di due numeri

Dopo aver chiesto all'utente di inserire 2 numeri (numero1 e numero2), il programma segnala se i numeri sono uguali. Se  $\text{numero1} > \text{numero2}$  e se  $\text{numero1} < \text{numero2}$ .

Come prima cosa, inizializza le variabili numero1 e numero2 a 0.

Poi bisogna chiedere all'utente di inserire un numero e salvare la sua risposta dentro la variabile numero1.

Ripeti lo stesso procedimento per la variabile numero2.

Ora confronta i due numeri inseriti:

- $\text{numero1} = \text{numero2}$
- $\text{numero1} < \text{numero2}$
- $\text{numero1} > \text{numero2}$

Esegui il programma per controllarne il funzionamento.

### DISEGNIAMO CON SCRATCH · L'editor grafico

Oltre i Tab **Codice** e quello **Suoni** in Scratch esiste un terzo Tab chiamato **Costumi**. All'interno di quest'area puoi gestire la parte grafica del tuo programma, ovvero gli Sprite.

Puoi disegnare uno Sprite attraverso l'editor oppure modificarne uno precaricato.

Hai a disposizione diversi strumenti per poter disegnare uno Sprite: forme geometriche, pennelli, strumenti, secchiello e una tavolozza colore.

Ogni Sprite può avere più costumi. Per spiegarti cosa sono, la via migliore è farti un esempio.

Un cartone animato viene creato attraverso una sequenza di immagini che sembrano tutte uguali ma che in realtà differiscono per piccoli particolari. Quando vengono riprodotte una dopo l'altra a velocità costante, si ha l'impressione che il disegno si animi.

In Scratch vale lo stesso principio. I costumi rappresentano le immagini, se vengono riprodotte in sequenza generano la percezione di movimento.

Se uno Sprite ha un solo costume non potrà essere animato mentre se ne possiede più di uno allora lo vedremo muoversi liberamente.

Ad esempio il pipistrello ha 4 differenti costumi chiamati Bat-a, Bat-b, Bat-c e Bat-d. Puoi duplicarli cliccandoci sopra con il tasto destro del mouse o rinominarli scrivendo il nuovo nome nell'area di testo in alto a sinistra dell'editor grafico. Inoltre, attraverso gli strumenti a disposizione potrai modificare questi costumi a piacimento e crearne nuovi cambiando forma, colore, dimensione o posizione.

## I CLONI

In alcuni casi o in alcuni giochi, è necessario riprodurre più oggetti copie (figli) di uno originale (padre) per creare un particolare effetto durante la riproduzione come, ad esempio per rappresentare dei fuochi d'artificio o un'intensa nevicata. Gli oggetti creati a partire da un originale prendono il nome di **cloni** (possono essere attivi un massimo di 300 cloni nello stesso momento) e questi ereditano dall'originale script, costumi, suoni e proprietà.

### ATTIVITÀ GUIDATA · Un campo pieno di farfalle

*Elementi della scena*

Sprite: **1**

Sfondo: **1**

Categoria di codici utilizzati: **Controllo, Operatori, Aspetto, Variabili, Situazioni, Movimento**

Cosa c'è di più bello che vedere un prato fiorito pieno di farfalle che svolazzano? Per realizzare questa immagine, bisogna utilizzare il comando **crea clone** dalla categoria **Variabili** di Scratch.

Fissiamo per primo lo script che servirà a clonare l'oggetto originale, ossia lo script per il padre. Possiamo effettuare diverse scelte, ma in questo caso la farfalla si sdoppierà (figli) quando prenderemo la barra spaziatrice sulla tastiera.

Occupiamoci poi dello script relativo ai figli clonati. Trascina il blocco **Quando vengo clonato** dalla categoria **Controllo** nell'area script in modo che tutti i blocchi che attaccherai sotto verranno eseguiti solo dai cloni.

Usando il blocco **"cambia effetto colore di..."** e **"ruota di..."** potrai creare tante farfalline diverse partendo da un unico Sprite: infatti grazie a questi comandi, ogni volta che viene creato un clone questo cambierà colore e ruoterà di 30° per poi eliminarsi dopo un numero n di volte per non sovraccaricare eccessivamente il processore **"elimina questo clone"**.

All'interno del ciclo iterativo "**ripeti**", inserisci un blocco di ciclo per ripetere all'infinito lo **scivolamento** dello Sprite, impostando un numero casuale per il valore della coordinata x e uno per il valore della coordinata y.

Esegui il programma e premi la barra spaziatrice più volte per animare il tuo campo fiorito.