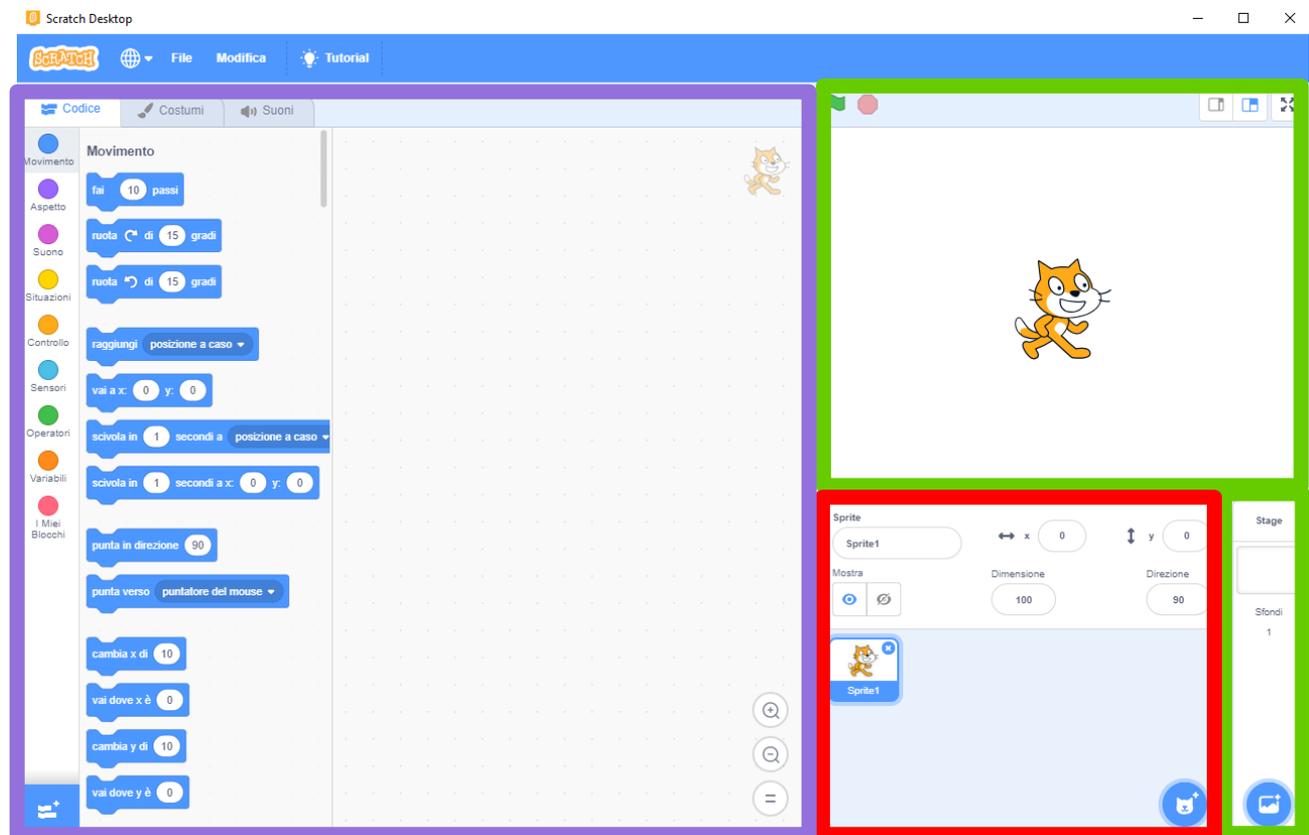


Teacherdojo 2019
MATERIALI PER I DOCENTI
Giorgia Bissoli, Alberto Montresor

Come si presenta Scratch

Ecco come appare la finestra di Scratch dopo averlo aperto sul proprio pc.
Per prima cosa, se la lingua è diversa dall'italiano, premete il tasto con il mappamondo in alto a sinistra e impostate l'italiano.

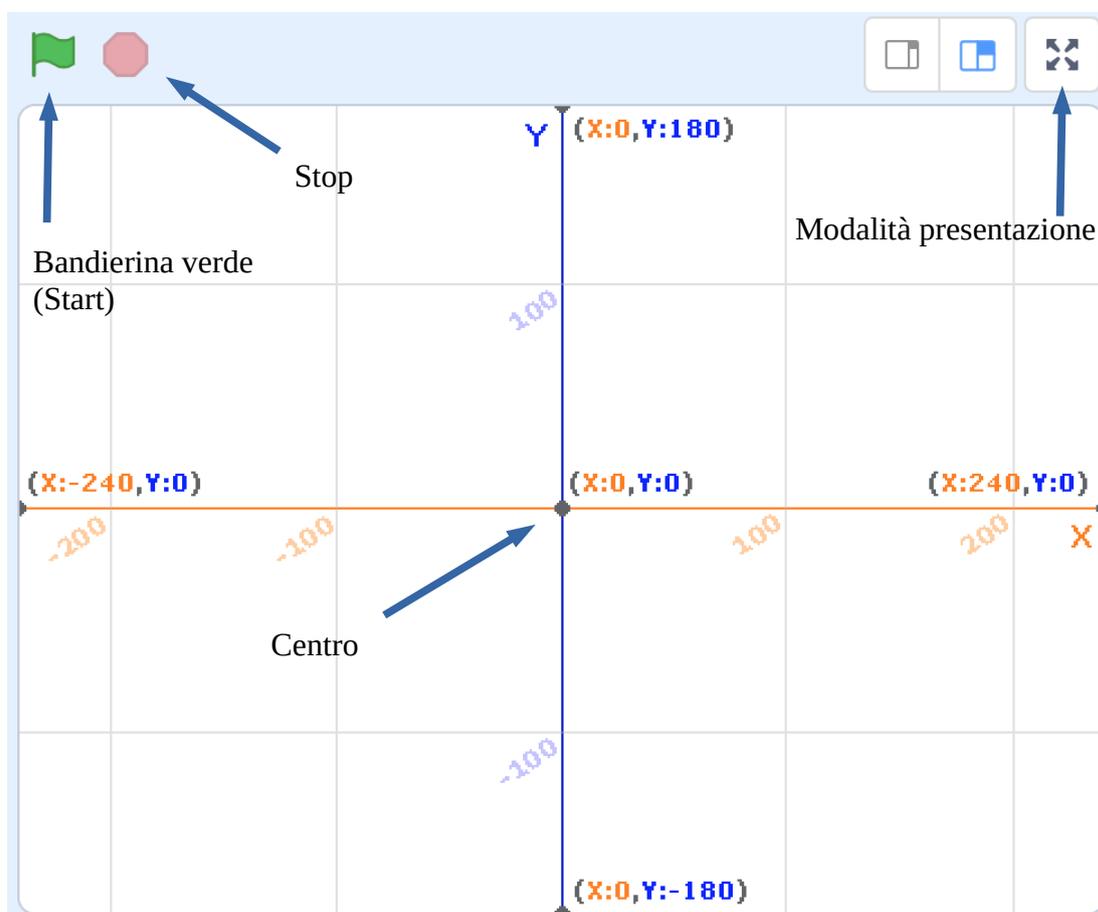


Gli elementi principali di Scratch sono 3: *stage*, *sprite* e *codice*.

L'area che vedete contornata in verde è la zona relativa agli **stage**, ossia gli sfondi presenti all'interno della nostra animazione.

- Nell'area rossa troviamo gli **sprite**, cioè i personaggi della nostra storia; lo sprite predefinito all'apertura di un nuovo programma è il gatto Scratch all'apertura di un nuovo programma.
- Infine all'interno dell'area viola si trova il **codice**, detto anche **script**, cioè il copione della nostra storia, che dice a personaggi e sfondi cosa fare e quando farlo

Gli stage



Come potete vedere nell'immagine, lo stage è organizzato secondo un sistema di coordinate cartesiane:

- 480 unità (240 positive e 240 negative) di larghezza
- 360 unità di altezza (180 positive e 180 negative).

Il **centro** dello stage si trova alle coordinate $x=0$ e $y=0$, x è uguale a 0 lungo tutta la linea arancione e $y=0$ lungo tutta la linea blu. Questo sistema di coordinate vi servirà per dire ai personaggi in che punto dello sfondo comparire o muoversi.

Per creare un nuovo sfondo dovete premere sull'icona in basso a destra dello schermo.

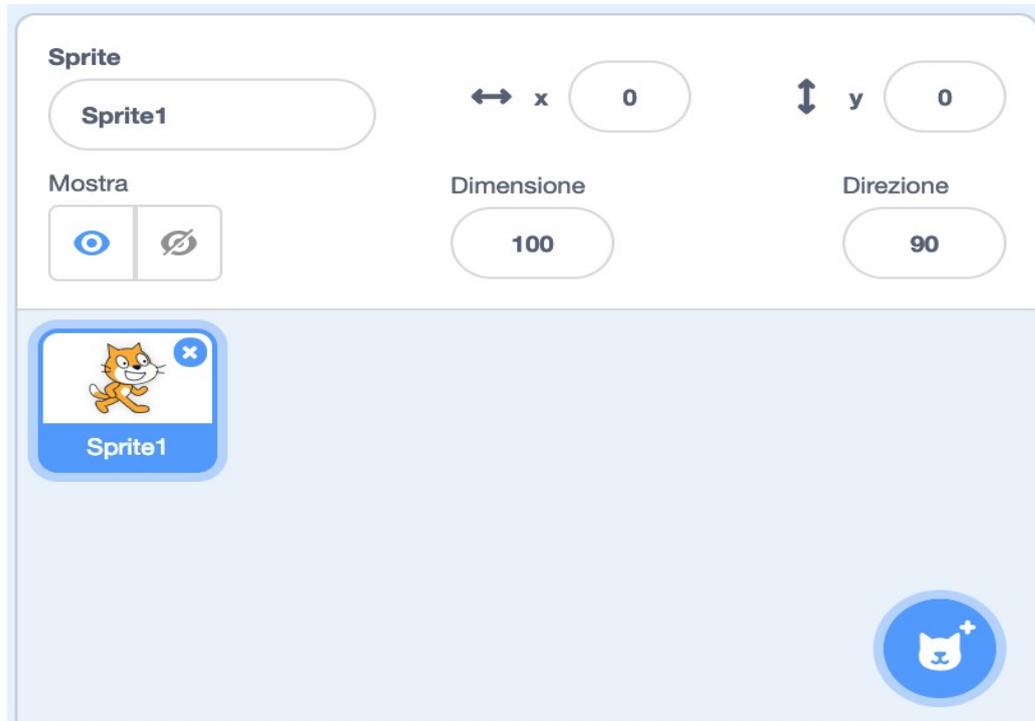
Sarà possibile creare un nuovo sfondo in diversi modi (dall'alto verso il basso) :

- caricarlo dal vostro pc
- farlo caricare a Scratch in modo casuale
- disegnarlo
- sceglierlo dalla libreria di Scratch

Provate a creare degli sfondi nei diversi modi!



Gli sprite



Nella sezione relativa agli sprite ci sono diverse informazioni relative agli stessi:

- Posizione x e y : provate a trascinare il gatto e spostarlo in un altro punto dello schermo, le coordinate del gatto si aggiorneranno con la nuova posizione.
- Dimensione dello sprite: provate a modificare la dimensione dello sprite e guardate cosa succede
- Direzione dello sprite: sarà spiegata nel dettaglio successivamente.

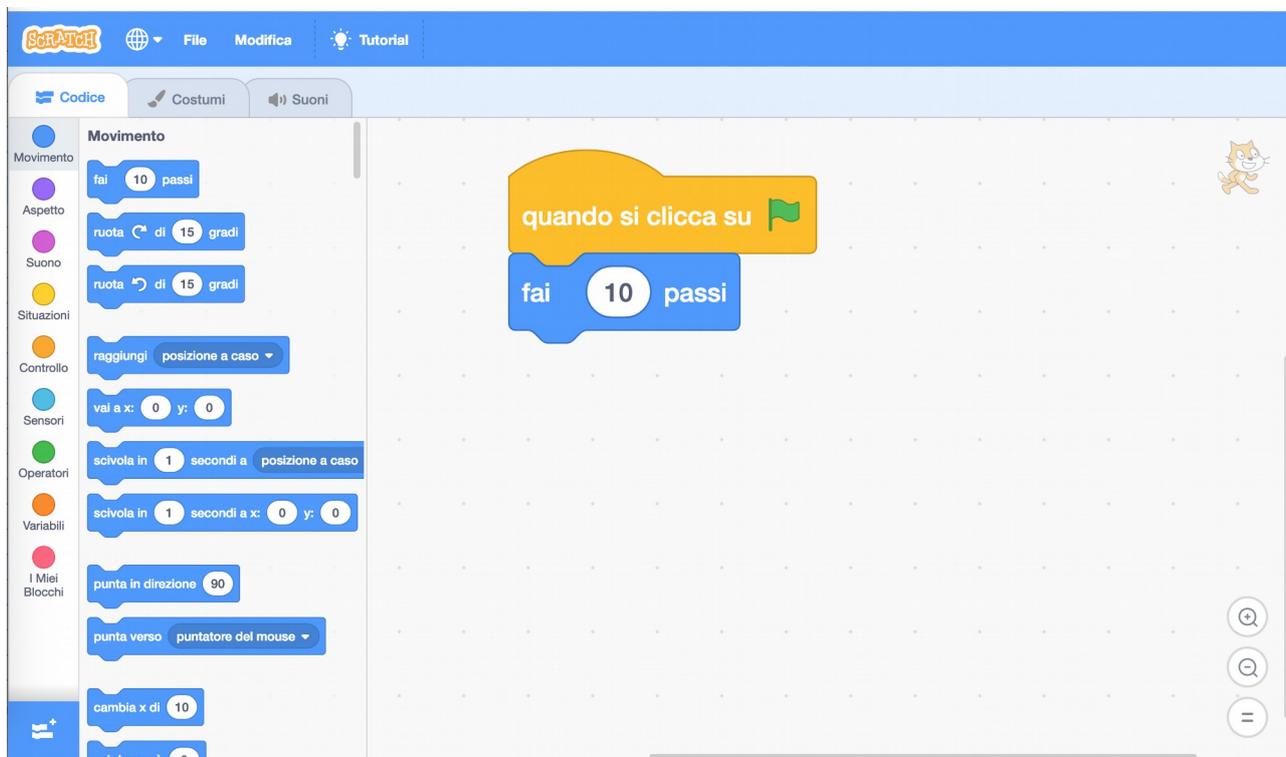
Per creare un nuovo sprite dovete premere sull'icona con la faccia del gatto disegnata. Sarà possibile creare un nuovo sprite in diversi modi (dall'alto verso il basso) :

- caricarlo dal vostro pc
- farlo caricare a Scratch in maniera casuale
- disegnarlo
- sceglierlo dalla libreria di Scratch

Provate a creare degli sprite nei diversi modi!



Il codice (script)



Lo spazio bianco al centro dello schermo e a sinistra dello stage è la porzione di schermo che deve contenere il codice che servirà per dare indicazioni precise al nostro personaggio su quello che dovrà fare.

Ogni personaggio in Scratch possiede un suo script personale.

Il personaggio eseguirà tutto quello che gli diremo di fare, senza se e senza ma, quindi nel caso non riuscissimo a fargli completare il compito, dipenderà interamente dal codice che è stato scritto.

Lo script è composto da una serie di blocchi colorati (a sinistra dello schermo) inseriti sequenzialmente uno dopo l'altro. Il personaggio eseguirà i comandi in ordine, partendo dal blocchetto più alto al più basso. Per comporre il vostro script dovete prendere i blocchetti dalla parte sinistra del foglio e trascinarli nella zona bianca centrale.

Affinché Scratch riesca a leggere correttamente lo script è necessario che i blocchetti si incastrino l'uno con l'altro proprio come i pezzi di un puzzle.



Provate a comporre lo script nell'immagine qui sopra in maniera corretta.

Ora premete la bandierina verde sul vostro schermo e osservate cosa succede. Fatelo ancora e ancora.

Proviamo a capire quello che è successo.

Lo script è composto da due istruzioni:

- **[quando si clicca sulla bandierina verde]** (Situazioni): significa che le successive istruzioni verranno eseguite quando premerete la bandierina verde presente sullo schermo con il mouse
- **[fai 10 passi]** (Movimento) che significa che il gatto si deve muovere di 10 unità sullo stage nella direzione in cui sta guardando.

Un'istruzione fa sì che uno sprite esegua un'azione, quindi le istruzioni devono essere chiare e non ambigue. Quando un programma viene eseguito vengono svolte le azioni specificate nelle istruzioni.

Guida agli esercizi

Le seguenti dispense sono state pensate per accompagnarvi nell'apprendimento dell'utilizzo del software. La metodologia a cui si fa riferimento è quella del problem-base-learning, durante le dispense infatti verranno proposti dei task da completare di difficoltà crescente.

Quello che si suggerisce di fare è di procedere, per ogni compito, come segue:

1. leggere il testo
2. pensare a come risolvere il problema
3. scomporre la risoluzione del problema in step
4. riscrivere i vari step della soluzione con il linguaggio di Scratch
5. se non funziona, cercate di capire in quale step della catena si è fermata la risoluzione del problema e cercate di risolvere il problema allo stesso modo del punto 2.

A fianco della scritta "Compito x" troverete dei pallini colorati che vi suggeriranno il colore dei blocchetti che dovrete utilizzare per riuscire a completare l'esercizio.

-  I blocchi **Movimento** includono tutte le istruzioni che è possibile impartire ad uno sprite per consentirgli di muoversi sulla scena.
-  I blocchi **Aspetto** includono tutte le istruzioni per consentire ad uno sprite di parlare e pensare (con il meccanismo dei fumetti, quindi visualizzando del testo), di apparire, scomparire e cambiare costume.
-  I blocchi **Suono** forniscono le istruzioni che è possibile inserire in un programma per riprodurre dei suoni (e.g. voce, musica, effetti).
-  I blocchi **Situazioni** forniscono le istruzioni per avviare uno script, per avviare tutti gli script, per inviare un messaggio (ad un altro script) e per gestire alcuni eventi quali, ad esempio, la rumorosità.
-  I blocchi **Controllo** includono le istruzioni per permettere ad uno sprite di ripetere più volte lo stesso comportamento, di clonarsi, di fermarsi (fermare l'esecuzione dello script) di attendere degli eventi specifici, etc.
-  I blocchi **Sensori** includono le istruzioni per percepire gli eventi esterni (clic del mouse, pressione su tastiera, movimento nella web cam) e interni relativi all'esecuzione di un programma e consentire a uno sprite di reagire in maniera specifica a un singolo evento.
-  I blocchi **Operatori** includono le istruzioni per eseguire operazioni aritmetiche e logiche, di valutare condizioni, di manipolare testi, etc.
-  I blocchi **Variabili** e liste includono tutte le istruzioni per creare nuove variabili/liste e per utilizzarle all'interno dei vostri programmi. Le variabili sono dei contenitori residenti in memoria (del computer) che servono a conservare e manipolare i dati (e.g. numeri, caratteri, parole, frasi, etc.). Le liste permettono di gestire più variabili correlate tra loro.

Tutti i compiti sono raccolti all'interno della galleria Scratch presente al seguente link:
<https://scratch.mit.edu/studios/25138653/>

Compito 1 – Crea un acquario!



Scegliete uno sfondo adeguato e inserite 1 pesce, 1 stella marina e 1 medusa.

Ogni volta che si clicca sulla bandierina verde:

- 1. Il pesce si muove, facendo 15 passi e ruotando di 15 gradi in senso orario.*
- 2. La stella marina compare al centro e scivola in 1 secondo in una posizione casuale dello schermo*
- 3. La medusa si muove prima verso l'alto di 200 unità e poi verso il basso di 200 unità*

(<https://tinyurl.com/teacherdojo1>)

- 1. Il pesce si muove, facendo 100 passi e ruotano di 90 gradi in senso orari.*



Il programma di default imposta dei valori che non sono quelli che io desidero. Come faccio a modificare i valori dei passi e dei gradi per “personalizzarli”?

In Scratch, i blocchetti sono composti da:

- parti di istruzioni generali [fai .. passi] che non possono essere modificate
- da parti di istruzioni relative al valore (parti in bianco) che invece possono essere modificate in base alle vostre necessità.

Provate a fare doppio click sulla parte bianca del blocchetto per modificare il valore dei passi e dei gradi.

Il vostro script dovrebbe comparire così adesso.

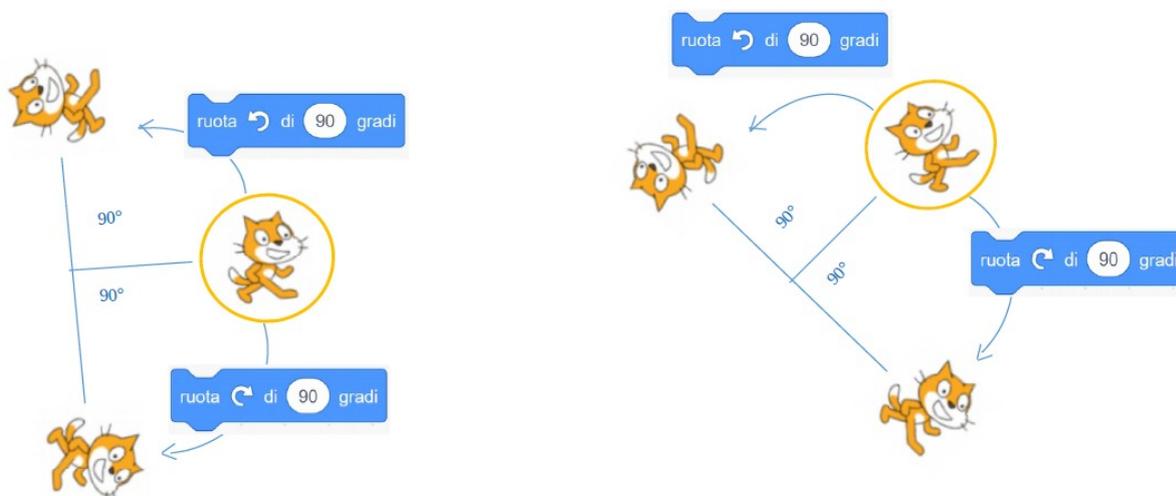


Un concetto importante: la direzione

Esistono due modi per cambiare la direzione dello sprite:

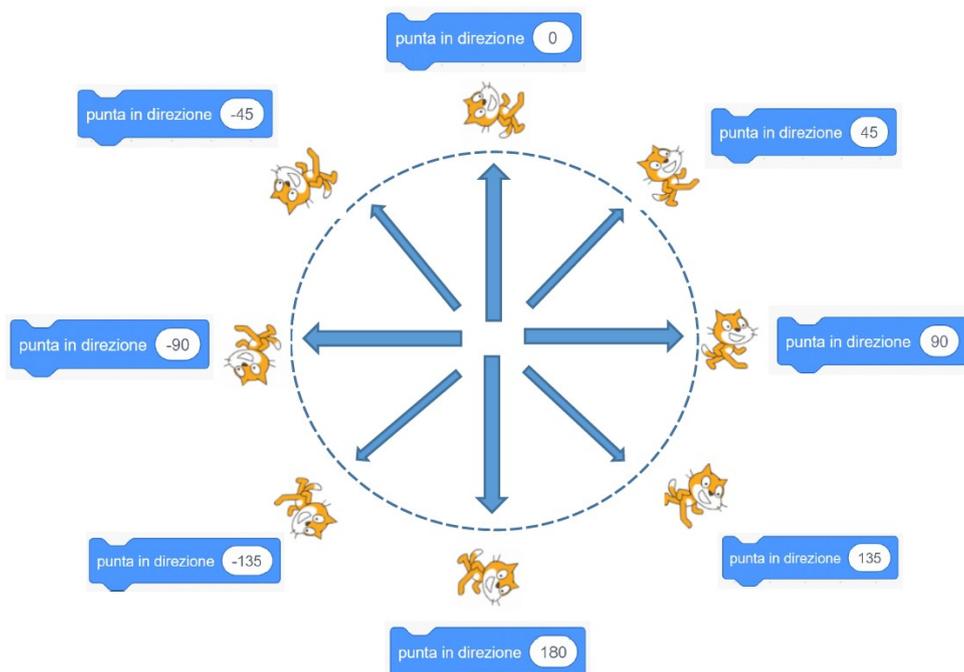
1. [ruota di ... gradi] (Movimento) - Rotazione relativa

Con questo blocco potrete decidere di far ruotare il gatto di n gradi in senso orario oppure in senso antiorario (deciso dalla freccia bianca). Fate attenzione, perché il gatto ruoterà di n gradi rispetto alla sua posizione iniziale di partenza e non in maniera assoluta.



2. [punta in direzione ..] (Movimento) - Rotazione assoluta

Indipendentemente dalla sua posizione iniziale il gatto punterà nella posizione indicata.



2. *La stella marina compare al centro e scivola in 1 secondo in una posizione casuale dello schermo*



Come si può far ripartire lo sprite dal centro dello schermo?

Utilizzate il blocco `[vai a x: .. y: ..]` (Movimento)

Utilizzando il blocchetto `[scivola in .. secondi a x: .. e y: ..]` (Movimento) il vostro sprite si sposterà da una posizione all'altra nel tempo da voi indicato: modificando il numero di secondi, modificherete la velocità di spostamento del gatto.

A differenza del blocchetto `[vai a x: .. y: ..]` (Movimento), in cui lo sprite compare da una posizione all'altra in brevissimo tempo, questo blocco permette di vedere lo spostamento da un punto all'altro dello stage.

Concetti importanti: lo stato e l'inizializzazione

Se utilizzate gli script scritti precedentemente, lo sprite si muove e ruota, ma ogni volta concluso lo script si trova in una nuova posizione, la stessa posizione da cui ricomincerà a muoversi la seconda volta che cliccherete sulla bandierina verde. In altre parole, l'effetto dell'esecuzione dello script dipende dalla posizione in cui si trova il gatto.

Per fare in modo che il gatto riparta sempre dalla stessa posizione, in questo caso il centro dello schermo, è necessario inserire all'inizio del codice la posizione in cui dovrà trovarsi una volta che si è premuto sulla bandierina verde, utilizzando il sistema di riferimento x, y introdotto precedentemente nella sezione "stage".

Ogni sprite è caratterizzato da uno **stato**, ovvero un insieme di valori che descrivono la sua posizione, la sua direzione, la sua dimensione, il suo essere visibili oppure no. Al termine di un'esecuzione, lo sprite conserva tutti i valori che aveva al termine della stessa.

Se all'inizio volete che lo sprite assuma una posizione particolare, una dimensione, dovete aggiungere dei blocchi di **inizializzazione**: ovvero dei blocchi che descrivono lo stato dello sprite all'inizio dell'animazione.

3. *La medusa si muove prima verso l'alto di 200 unità e poi verso il basso di 200 unità*

Un concetto importante: la traslazione

Utilizzando le coordinate cartesiane, è possibile

Come già detto precedentemente, per gli spostamenti all'interno dello stage si utilizza un sistema di riferimento cartesiano. Come per la rotazione, esistono due modi per far traslare uno sprite:

1. **[cambia y di ..]** (Movimento) - Movimento relativo

Con questo blocco potrete decidere di far spostare il gatto di n unità verso l'alto (numero positivo) o verso il basso (numero negativo). Fate attenzione, perché il gatto si sposterà di n unità rispetto alla sua posizione di partenza e non in maniera assoluta.

2. **[vai dove y è ..]** (Movimento) - Movimento assoluto

Il gatto, pur mantenendo il valore assegnato alla coordinata x , si sposterà sulla posizione y indicata nei blocchetti indipendentemente dal valore y iniziale.

La stessa cosa avviene con i blocchetti **[cambia x di ..]** e **[vai dove x è ..]**, ma in questo caso le traslazioni avvengono in orizzontale.



Cosa accade se inserite un numero negativo nella finestra bianca dei blocchi?

I numeri negativi fanno muovere lo sprite nella direzione opposta rispetto ai numeri positivi. Se lo sprite sta guardando verso destra, un numero positivo lo farà muovere verso destra, mentre un numero negativo lo farà muovere verso sinistra. Viceversa, se lo sprite sta guardando verso sinistra, un numero positivo lo farà muovere verso sinistra mentre un numero negativo lo farà muovere verso destra.



Nonostante io abbia inserito i blocchetti il mio sprite sembra fermo, perché?

Anche se avete inserito tutti i blocchetti della sezione movimento corretti il vostro sprite non sembra muoversi dalla posizione iniziale. In realtà lo sprite sta eseguendo tutto il vostro script e lo sta mandando in esecuzione molto velocemente, tanto che vi sembra fermo! Utilizzando il blocco **[attendi .. secondi]** (Controllo) tra un blocco e l'altro potrete "rallentare" il vostro script. Prima di ogni movimento, il vostro sprite aspetterà un secondo, in questo modo l'effetto sarà totalmente diverso.



Un concetto importante: la ripetizione

Potete accorciare il vostro script utilizzando il blocco **[ripeti .. volte]** (Controllo). Il blocco sembra una “C”, il gatto ripeterà n volte tutto quello contenuto all’interno della “C”. Quello che appare prima o dopo la C non viene ripetuto, ma eseguito una volta sola.



Un attimo per giocare!

Utilizzando i blocchi che abbiamo incontrato finora, prendetevi un attimo per giocare!

- Provate a realizzare piccole animazioni: scegliete un sfondo e muovete lo sprite attraverso di esso
- Adesso che conoscete le ripetizioni, utilizzatele per “rallentare” lo sprite: invece di muovere lo sprite di 100 passi, muovete lo sprite 10 volte di 10 passi: il movimento sarà più fluido.
- Sbizzarrite la fantasia!

Compito 2 – Fai muovere per sempre i pesci nell’acquario!



Quando si clicca sulla bandierina verde, il pesce deve comparire al centro dello schermo e continua ad andare per sempre da una parte all’altra dello schermo facendo 10 passi alla volta.

<https://tinyurl.com/teacherdojo2>

 Come posso far andare avanti lo sprite per sempre?

Provate a trovare una soluzione da soli. Poi leggete la pagina successiva.

Un concetto importante: la ripetizione infinita

Per ripetere un insieme di blocchi all’infinito, possiamo utilizzare il blocco **[per sempre]** (Controllo): a differenza del blocco **[ripeti .. volte]** (Controllo) questo blocco non ha l’incastro sul fondo, quindi nessun blocco può essere posizionato dopo di questo.

Tutto quello contenuto all’interno del blocchetto viene ripetuto per sempre e quindi eventuali blocchetti successivi non verrebbero mai mandati in esecuzione. Come per il **[ripeti .. volte]** (Controllo) il blocco **[per sempre]** (Controllo) ripete solo i blocchetti contenuti all’interno della C e non quelli precedenti.

Nel caso specifico, se volete che lo sprite compaia al centro dello schermo ogni volta che premete sulla bandierina verde e che continui ad andare da una parte all’altre dello schermo per sempre, dovrete inserire all’interno del per sempre solo i blocchi relativi al movimento del gatto e non quello sulla sua posizione iniziale. Provate a inserire **[vai a x: .. y: ..]** (Movimento) all’interno del ciclo per sempre e guardate cosa succede!



 Come faccio a non far scomparire lo sprite oltre il bordo dello stage?

Per evitare che il gatto esca dallo schermo dovete inserire il blocchetto **[rimbalza quando tocchi il bordo]** (Movimento). Lo sprite rileva il bordo dello stage come un confine impossibile da sorpassare, di conseguenza torna indietro ogni volta che raggiunge uno dei bordi dello stage.

 Come faccio a non far capovolgere lo sprite quando tocca il bordo?

Per evitare che lo sprite si capovolga a testa in giù dovete usare il blocchetto **[usa stile rotazione destra – sinistra]** (Movimento)

In questo modo il vostro gatto guarderà a destra quando si sposterà a destra, mentre guarderà a sinistra quando si sposterà verso sinistra.

A destra del blocchetto compare una freccina bianca rivolta verso il basso. Premete sulla freccina, il programma vi darà altre opzioni disponibili da selezionare. Provate a modificare lo script inserendo, uno alla volta, tutti e tre stili di rotazione e guardate cosa succede.



Compito 3 – Anima il tuo nome!



Scegli nella libreria degli sprite le lettere che compongono il tuo nome.

Lettera 1 : quando si clicca sulla lettera 1, la lettera sparisce e dopo 3 secondi ricompare.

Lettera 2: quando si clicca sulla lettera 2, la lettera 2 cambia colore.

Lettera 3: quando si clicca sulla lettera 3, cambia dimensione.

Lettera.....

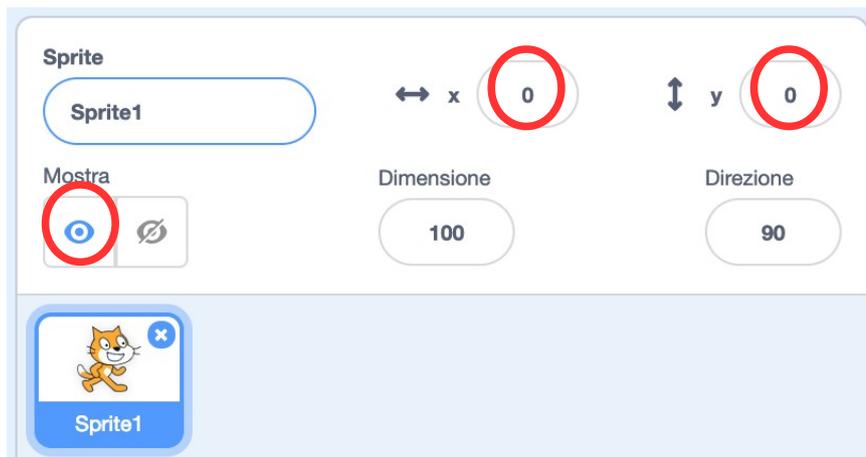
[<https://tinyurl.com/teacherdojo3>]

Lettera 1 : quando si clicca sulla lettera 1, la lettera sparisce e dopo 2 secondi ricompare.

Con il blocchetto **[nascondi]** (Aspetto) potete far sparire lo sprite di scena.

 Il mio sprite è sparito; come posso farlo ricomparire?

A volte lo sprite scompare perché l'avete nascosto e vi siete dimenticati di farlo riapparire; a volte scompare perché esce dallo schermo. Se siete in questa situazione, potete farlo ricomparire tramite interfaccia, assicurandovi di cliccare sull'occhio aperto e di metterlo in posizione 0,0.



Lettera 2: quando si clicca sulla lettera 2, la lettera 2 cambia colore.

Lettera 3: quando si clicca sulla lettera 3, cambia dimensione.



Come faccio ad ingrandire e far cambiare il colore agli sprite?

Per cambiare dimensione allo sprite utilizzando i blocchetti nello script ci sono due modi:

1. **[cambia dimensione di ..]** (Aspetto) - **cambio di dimensione relativo**

Con questo blocco potete decidere di far cambiare la dimensione dello sprite di n unità rispetto alla dimensione che lo sprite ha in quel momento. Fate attenzione, perché lo sprite cambierà la sua dimensione rispetto alla sua dimensione precedente e non in maniera assoluta.

2. **[porta dimensione al ..%]** (Aspetto) - **cambio di dimensione assoluto**

Indipendentemente dalla grandezza iniziale, lo sprite diventa della dimensione indicata; la dimensione finale dello sprite è data in % rispetto alla dimensione predefinita. Portando la dimensione dello sprite al 50% la lettera diventa la metà, portando la dimensione al 200 % invece la lettera diventa il doppio del gatto caricato inizialmente da Scratch.

Per modificare il colore dello sprite, vale lo stesso principio:

1. **[cambia effetto colore di ..]** (Aspetto) - **Cambio di colore relativo**

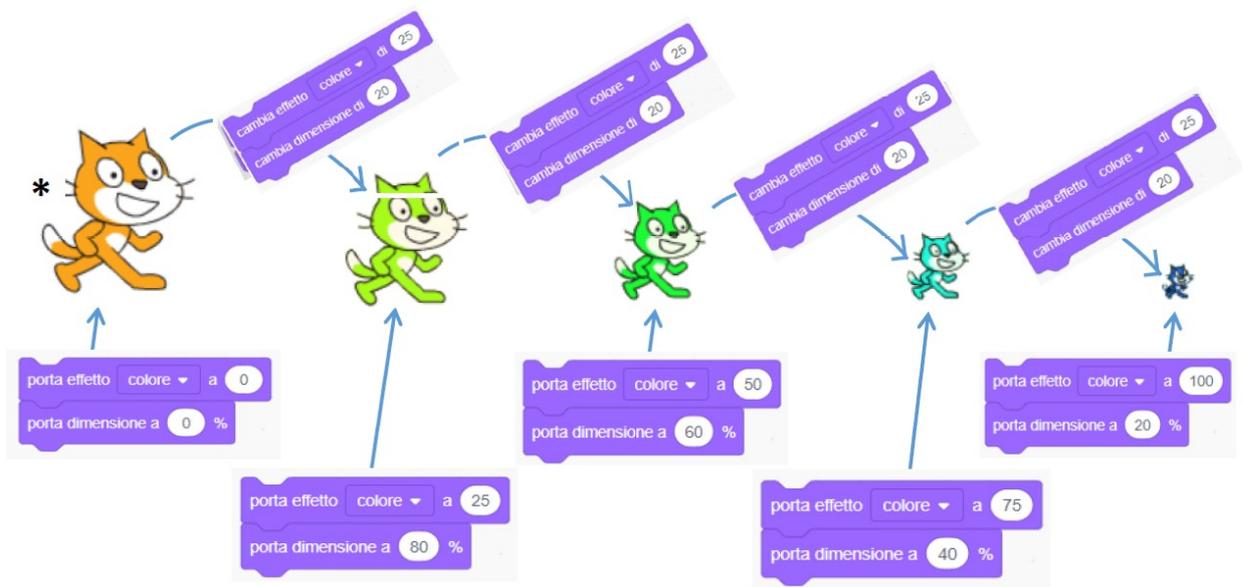
Con questo blocco potete cambiare il colore dello sprite di n unità rispetto al colore precedente dello sprite, quindi il colore non cambierà in maniera assoluta.

2. **[porta effetto colore a ..]** (Aspetto) - **Cambio assoluto**

Indipendentemente dal colore iniziale, lo sprite diventa del colore indicato.

 Una volta mandato in esecuzione lo script, lo sprite non torna alla sua configurazione iniziale. Come posso farlo tornare all'aspetto di partenza?

Anche in questo caso, come per il mostra e nascondi, una volta mandato in esecuzione il blocchetto lo sprite rimane dell'aspetto indicato. In questo caso, dovete dire allo sprite come apparire (dimensione al 100% e colore arancione) appena si clicca sulla bandierina.



porta effetto colore a 50

- ✓ colore
- fish-eye
- mulinello
- effetto pixel
- mosaico
- luminosità
- fantasma

Prova a modificare il valore degli effetti speciali!

La stessa cosa vale per gli altri effetti.

Un attimo per giocare!

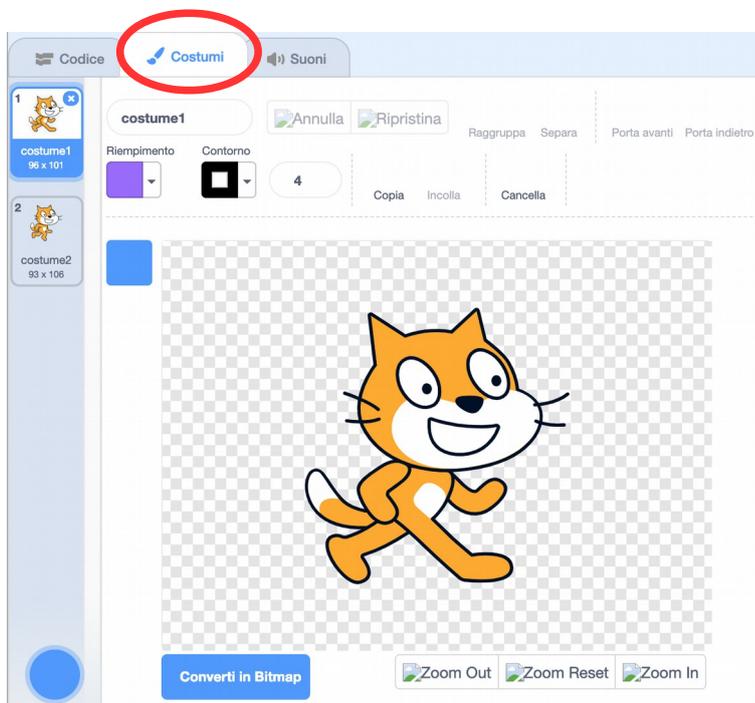
Utilizzando i blocchi che abbiamo incontrato finora, prendetevi un attimo per giocare!

- Provate a completare l'animazione del vostro nome scegliendo i diversi effetti.
- Ricordate che avete a disposizione anche le ripetizioni!
- Sbizzarrite la fantasia!

Un concetto importante: i costumi

Tutte le animazioni create fino a qui non tengono conto di un elemento essenziale: anche se fino ad ora avete imparato, per esempio, a far muovere il gatto (spostandosi in un altro punto dello schermo), la posizione delle gambe del gatto è rimasta invariata. Un'animazione più verosimile dovrebbe invece mostrare anche una modificazione dell'aspetto dello sprite. Nei cartoni animati per esempio, questo effetto è dato da una serie di fotogrammi che vengono mostrati in sequenza molto velocemente: per mostrare che un personaggio sta camminando, si disegnano diverse immagini del personaggio con braccia e gambe in posizioni diverse. Allo stesso modo, in Scratch gli sprite possono avere diversi "costumi" da indossare, che sono l'equivalente dei fotogrammi appena citati.

Cliccando sulla scritta "costumi" in alto a sinistra del foglio potrete vedere i costumi che ha a disposizione il vostro sprite, disegnarne o importarne di nuovi.



Nel caso specifico del gatto, sono presenti due costumi:

1. "costume1" : gatto con le gambe stese
2. "costume2" : gatto con le gambe piegate.

Quando viene eseguito uno script potete includere delle istruzioni che cambino i costumi utilizzando i blocchi

1. **[passa al costume ...]** (Aspetto) - **Cambio di costume assoluto**

Il gatto passerà al costume specificato indipendentemente dal costume di partenza

2. **[passa al costume seguente]** (Aspetto) - **Cambio di costume relativo**

Lo sprite passa al costume successivo, l'ordine dei costumi è dato dal numerino presente all'interno del quadrato di ogni costume (angolo in alto a sx).

Compito 4 – Crea una storia!



Prima di cominciare a programmare create due personaggi e due sfondi.

Quando si clicca sulla bandierina verde il gatto e l'altro personaggio dialogano tra loro:

personaggio 1 dice: "Ciao, come stai?"

personaggio 2: "Tutto bene grazie, e tu?"

personaggio 1: "Benissimo!"

Dopo che i due personaggi hanno finito di parlare, lo sfondo cambia.

(<https://tinyurl.com/teacherdojo4>)



Come posso far parlare uno sprite?

Utilizzando il blocco **[dire .. per 2 secondi]** (Aspetto) potete far dire qualcosa agli sprite proprio come avviene nei fumetti. La stessa cosa accade con il blocchetto **[pensa .. per 2 secondi]** (Aspetto), ma ovviamente il contorno del fumetto sarà leggermente diverso.



Come faccio a sincronizzare la conversazione tra i due personaggi?

Prima di tutto ricordatevi che ogni sprite ha il suo script, quindi per far parlare il gatto dovete cliccare sull'icona del gatto sotto lo stage e inserire le istruzioni che deve seguire, mentre per far parlare l'altro personaggio dovrete prima cliccare sull'altro personaggio e poi inserire le istruzioni nello spazio vuoto.

Un concetto importante: la sincronizzazione

Molto probabilmente, avete sincronizzato le vostre azioni utilizzando opportunamente delle attese. Questo funziona finché i programmi che state scrivendo sono semplici. Al crescere del numero di azioni, tuttavia, è facile fare confusione.

Per questo motivo, esiste due istruzioni specifiche per sincronizzare le azioni:

1. **[invia a tutti messaggio 1]** (Situazioni)

Si tratta di un blocchetto che serve per inviare un segnale a tutti gli altri sprite.

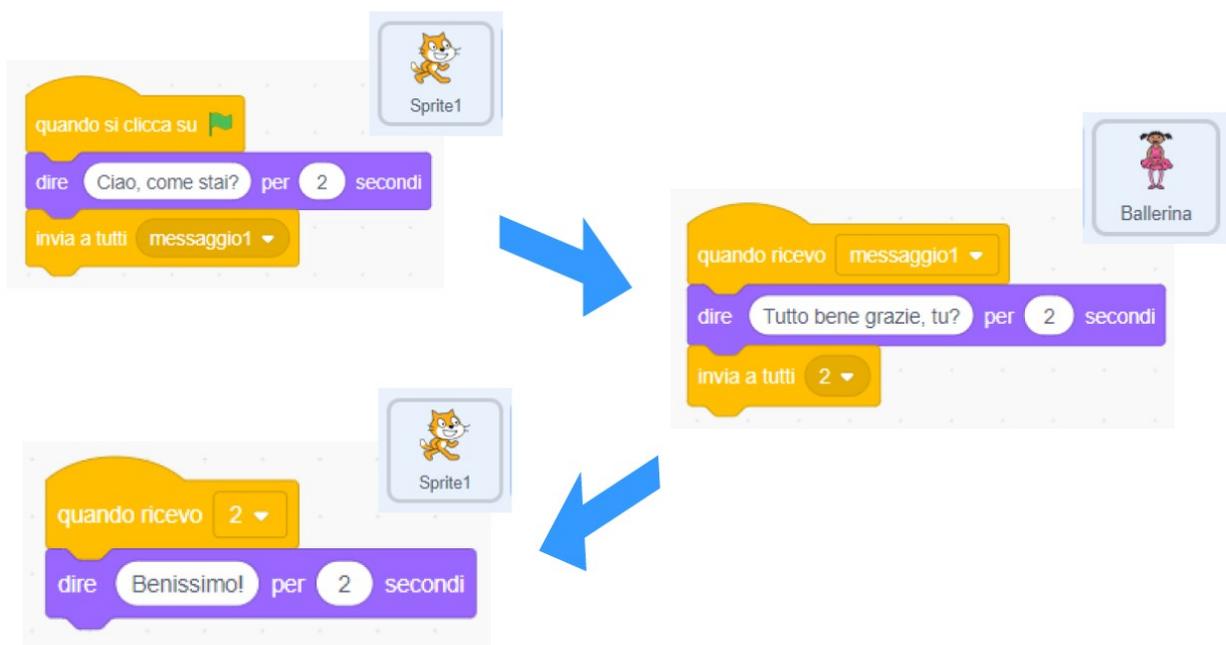
Il nome del messaggio viene scelto tra quelli disponibili nella finestra, oppure si possono creare nuovi messaggi.

2. **[quando ricevi messaggio 1]** (Situazioni)

Si tratta di un'istruzione di input, che serve invece per mandare in esecuzione delle istruzioni quando lo sprite riceve il segnale del “messaggio 1”.

Il messaggio viene inviato a tutti gli sprite, compreso il mittente stesso.

I messaggi sono fondamentali per sincronizzare istruzioni di aspetto, movimento o altro tra diversi sprite.



Dopo che i due personaggi hanno finito di parlare, lo sfondo cambia.

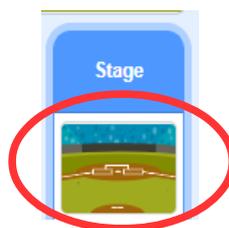
 Come faccio a cambiare lo sfondo?

Come inserire un nuovo sfondo è spiegato all'inizio delle dispense.

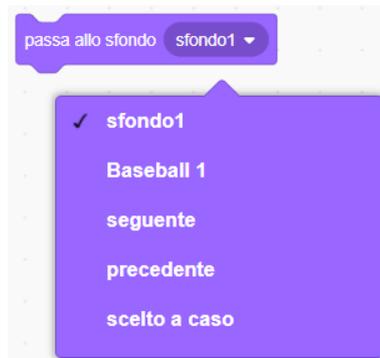
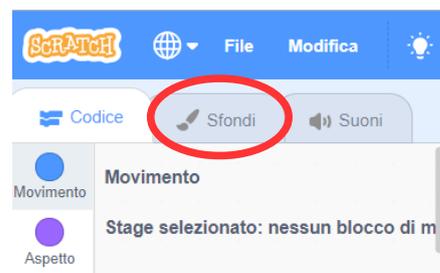
Per far cambiare sfondo all'interno dello script invece dovete utilizzare il blocco **[passa allo sfondo...]**. Cliccando sulla freccia bianca potrete selezionare lo sfondo da far comparire in base al nome.

Per visualizzare tutti gli sfondi disponibili e i loro nomi dovete:

1. cliccare sull'immagine sotto la scritta stage



2. cliccare sul tasto "sfondi" in alto a sinistra.



Un attimo per giocare!

Utilizzando i blocchi che abbiamo incontrato finora, prendetevi un attimo per giocare!

- Provate a completare l'animazione del vostro nome scegliendo i diversi effetti.
- Ricordate che avete a disposizione anche le ripetizioni!
- Sbizzarrite la fantasia!

Funzione penna!

Il cambio di sfondo può diventare anche un blocco di input!

Inserendo nello script il blocco **[quando lo sfondo passa a ...]** verranno mandate in esecuzione le istruzioni seguenti solo quando comparirà lo sfondo selezionato.

Con Scratch potete aggiungere delle estensioni tra cui la funzione penna. Cliccate nell'icona nell'angolo in basso a sinistra e selezionate il riquadro della "Penna", all'elenco dei vostri blocchetti dovrebbero essersi aggiunta l'icona con la penna.



Provate a rifare il compito 6, ma questa volta disegnando un quadrato.

 Come faccio a lasciare il segno quando il gatto si muove?



Inserendo l'estensione della funzione penna è come se aveste detto al vostro sprite di prendere in mano una penna. Per cominciare a lasciare il segno, lo sprite dovrà semplicemente "appoggiare" la penna al foglio con il blocchetto **[penna giù]**. Allo



stesso modo, per sollevare la penna dal foglio e non lasciare più alcun segno dovrete utilizzare il blocco **[penna su]**. Potete modificare il colore e la dimensione della penna utilizzando gli altri blocchetti.

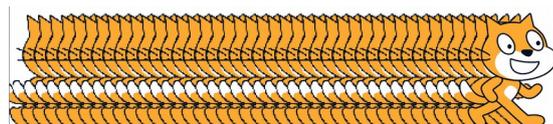
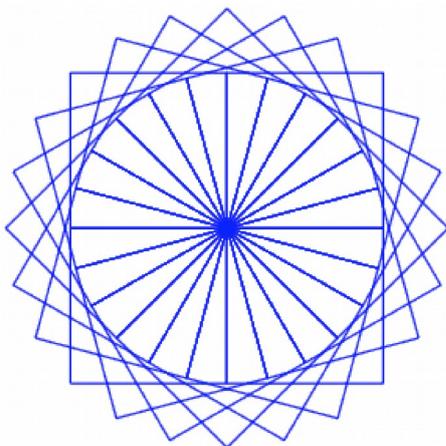
 Voglio ricominciare, come faccio a pulire il foglio?

Per pulire il foglio dovete inserire il blocco **[pulisci]**, tutto quello che è stato disegnato con la funzione penna verrà automaticamente cancellato.



Con il blocco **[timbra]** invece il vostro sprite verrà "impresso" sullo schermo.

Provate a disegnare qualche forma geometrica e qualche figura strana!



Ed ora in classe!

Qui di seguito faremo alcuni esempi di come sia possibile usare Scratch per creare alcune risorse didattiche molto semplici, con i pochi blocchi visti fino ad ora.

Link	Materia	Descrizione
https://scratch.mit.edu/projects/295670171/	Italiano	Creazione di un libro digitale (Illustrazioni, suoni e didascalie realizzate dai bambini seguendo un racconto).
https://scratch.mit.edu/projects/100094/editor/	Italiano	Narrazione a fumetti di un racconto di Gina Labriola "Il pianeta Faidatè" adattato e illustrato per il libro di testo "I colori dell'arcobaleno 2" ed. Il Capitello
https://scratch.mit.edu/projects/148481300/editor/	Storia dell'arte	Animazione di un quadro di Van Gogh.
https://scratch.mit.edu/projects/259324576/	Storia dell'arte	Omaggio a Matisse.
https://scratch.mit.edu/projects/218655878/	Geografia	Presentazione di una regione d'Italia: il Trentino Alto Adige.
https://scratch.mit.edu/projects/206736192/	Geografia	Mappa interattiva della Rimini.
https://scratch.mit.edu/projects/218658322/	Storia	La mummificazione degli Egizi.
https://scratch.mit.edu/projects/104306936/editor/	Inglese	Breve dialogo in lingua inglese.
https://scratch.mit.edu/projects/165329584/editor/	Matematica	Disegnare poligoni regolari.

Tutti i progetti sono raccolti nella seguente galleria: <https://scratch.mit.edu/studios/6109401/>

Compito 5 – Crea un quiz!



Crea un quiz!

(<https://tinyurl.com/teacherdojo5>)

Provate a capire come scomporre il compito. Creare un quiz significa:

1. Fare una domanda
2. Registrare la risposta
2. Controllare se la risposta è giusta o sbagliata
3. Dare un feedback (positivo o negativo) in base al tipo di risposta ricevuta

 Come faccio a fare una domanda e registrare la risposta?

L'istruzione **[chiedi ... e attendi]** permette di fare una domanda e di registrare la risposta dell'utente. La domanda va inserita all'interno dello spazio bianco. una volta che il blocco verrà eseguito comparirà nella parte inferiore dello schermo una barra in cui potrete scrivere la risposta. Per inviare la risposta dovete premere invio.



La risposta viene registrata automaticamente da Scratch all'interno del blocchetto **[risposta]**.

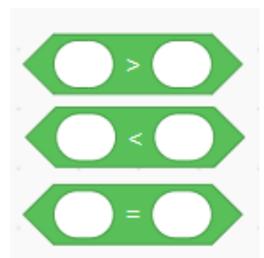
Attenzione: Scratch registra le risposte così come sono, tenendo conto di maiuscole, minuscole e spazi!

 Come faccio a controllare se la risposta è giusta o sbagliata?

Gli operatori di confronto vengono utilizzati per verificare una determinata condizione: all'interno delle due finestre bianche devono essere inseriti i valori che volete confrontare.

Esistono 3 operatori di confronto :

- **[... = ...]** è vero quando il valore della finestra a sinistra è equivalente al valore della finestra a destra, mentre è falso quando i due valori sono diversi.
- **[... > ...]** è vero solo se il valore nella finestra a sinistra è maggiore del valore nella finestra a destra.
- **[... < ...]** è vero solo se il valore nella finestra a sinistra è minore del valore nella finestra a destra.



 E se ci fosse più di una risposta giusta?

Per comporre nuove condizioni potete utilizzare gli operatori logici: all'interno dei due esagoni vanno inserite le condizioni da comporre.



Esistono 3 tipi di operatori logici:

- **[... e ...]** è vero solo se entrambe le condizioni sono vere.
- **[... o ...]** è falso solo se entrambe le condizioni sono false.
- **[non ...]** è vero se la condizione è falsa ed è falsa se la condizione è vera.

 Come faccio a dare un feedback (positivo o negativo) in base al tipo di risposta ricevuta?

Un concetto importante: l'istruzione condizionale

Il blocco di controllo che vedete qui in basso **[se...allora...altrimenti]** è un tipo di istruzione denominata condizionale perché la verifica di una determinata condizione è prioritaria all'esecuzione delle istruzioni contenute nel blocco. Ogni volta che il blocco va in esecuzione controlla se una determinata condizione si è verificata e decide come rispondere.



Nella finestra a forma di esagono va inserita la condizione che si intende verificare. Il blocco ha la forma di una "E": nella parte superiore della E dovete inserire le istruzioni da mandare in esecuzione se la condizione si è verificata, nella parte inferiore invece dovete inserire le istruzioni da mandare in esecuzione quando la condizione è falsa.

 Come faccio a ripetere la domanda fino a che la risposta non è corretta?

E' possibile fare in modo che la domanda venga ripetuta fino a quando non viene data la risposta giusta utilizzando il blocco **[ripeti fino a quando ...]**. Il blocco manderà in esecuzione determinate istruzioni fino a quando non si verificherà una determinata condizione.

Tutto quello contenuto all'interno della "C" viene ripetuto, a meno che la condizione inserita nell'esagono non sia vera.



Provate a creare un quiz con più domande e a segnare quante risposte corrette vengono date.



Come faccio a segnare un “punto” ogni volta che la risposta è corretta?

Un concetto importante – Le variabili

Uno dei concetti fondamentali della programmazione è quello di **variabile**. Una variabile è un dato che ha un **nome** e un **valore**. E’ attraverso le variabili che i programmi risolvono i problemi di calcolo.

Al contrario del foglio di calcolo, dove le variabili sono le celle della griglia, sono già disponibili e hanno nomi standard come A1, B7, nei programmi le variabili devono essere prima **dichiarate**.

- Il valore può essere letto in qualunque momento, utilizzando il **nome** come riferimento
- Il loro valore viene modificato con la scrittura diretta, ma attraverso altre operazioni:
 - assegnazione di un nuovo **valore** ad una variabile, utilizzando il **nome** come riferimento
 - incremento del **valore** di una variabile, utilizzando il **nome** come riferimento

Le istruzioni relative alle variabili si trovano nei blocchetti arancioni.



Per prima cosa cliccate su “crea una variabile” e datele un nome significativo, in questo caso può andar bene *Punti*. Create la variabile per tutti gli sprite. Nell’angolo in alto a sinistra dello stage dovrebbe essere comparso un rettangolino con il nome della variabile e il suo valore.

Per la gestione di un variabile ci sono due blocchi principali:

- **[porta .. a ..]** porta il valore della variabile selezionata al valore specificato.
- **[cambia .. di ..]** modifica il valore della variabile selezionata della quantità specificata.

Nel caso specifico, all’inizio del quiz dovrete inserire **[porta punti a 0]** (Variabili) mentre ogni volta che la risposta sarà corretta dovrete inserire **[cambia punti di 1]** (Variabili). In questo modo, alla fine del quiz, la variabile punti indicherà il numero totale delle risposte corrette.

Un esercizio difficile

Fate in modo che il vostro quiz accetti al massimo tre risposte sbagliate, e poi passi alla prossima domanda. Qualche suggerimento:

- create una variabile *Tentativi*
- prima di una domanda, portate *Tentativi* a 3
- dopo ogni risposta sbagliata, sommate -1 a *Tentativi*
- ripetete fino a quando $Tentativi = 0$ oppure la risposta è corretta

Compito 6



Prima di cominciare, crea 2 sprite.

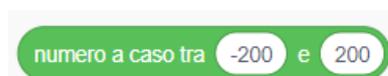
Quando si clicca sulla bandierina verde lo sprite 1 scivola in maniera casuale in diverse posizioni dello schermo. Quando lo sprite 1 tocca lo sprite 2, dice qualcosa per due secondi e il programma termina.

(<https://tinyurl.com/teacherdojo6>)



Come posso far muovere lo sprite in maniera casuale?

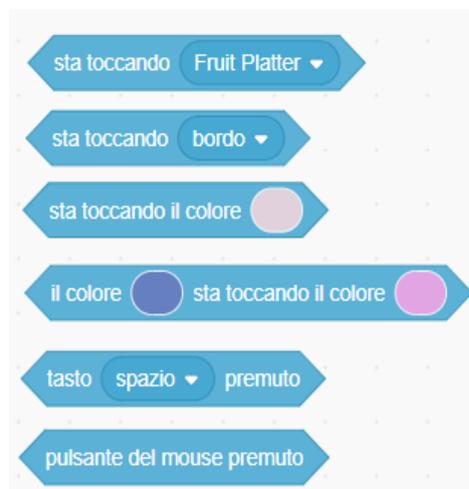
Utilizzando il blocco **numero a caso tra ... e ...** puoi definire un intervallo di valori entro cui il programma andrà a scegliere un numero intero in maniera casuale.



Come posso far capire allo sprite 1 che sta toccando lo sprite 2?

In Scratch ogni sprite ha dei sensori che possono percepire diverse situazioni. Nel caso specifico lo sprite 1 deve capire quando sta toccando lo sprite 2. Per fare ciò, sfruttate i blocchi azzurri per dire allo sprite cosa fare nel caso in cui la condizione **sta toccando ...** si verifichi.

Oltre ai sensori di tocco sono presenti anche i sensori per i colori e sensori per capire se i tasti del computer o del mouse vengono premuti.



Come faccio a fermare il gioco?

Utilizzando il blocco **ferma tutto** (Controllo) potrete interrompere tutti gli script di tutti gli Sprite.

Ed ora in classe!

http://favolealcomputer.weebly.com/	Laboratori di animazioni e narrazioni digitali (digital storytelling) in italiano e in altre lingue, utilizzando Scratch e il testo “Favole al Telefono” di Gianni Rodari.
https://www.codingcreativo.it/	Sul sito sono raccolti diversi materiali con relativi tutorial.
https://ltaonline.wordpress.com/2014/03/25/scratch-arte-e-scienze-al-romecup-2014-con-studenti-entusiasti/	Scratch, arte e scienze al RomeCup 2014
http://www.lafucinadelle scienze.it/wordpress/archives/2938	La fucina delle scienze Sistema preda e predatore
http://www.lafucinadelle scienze.it/wordpress/archives/2967	Rete ecologica
http://www.lafucinadelle scienze.it/wordpress/archives/3021	Fluido non newtoniano
http://www.lafucinadelle scienze.it/wordpress/archives/3042	Somma di vettori
http://www.lafucinadelle scienze.it/wordpress/archives/3065	Lo spettro sonoro
http://www.lafucinadelle scienze.it/wordpress/archives/3070	Il sistema solare

Crea un videogioco!

Realizziamo un videogioco in cui piove frutta dal cielo. Il giocatore deve riuscire a prendere le mele, muovendosi con le frecce direzionali del computer, per guadagnare i punti! Se la mela cade a terra il giocatore ha perso!

(<https://tinyurl.com/teacherdojo7>)

Proviamo a scomporre il compito:

1. Scendono mele dal cielo.
 1. La mela compare poco sotto bordo superiore dello schermo (in una posizione con coordinata x casuale) e si sposta lungo l'asse y fino ad arrivare al bordo inferiore dello schermo
 2. Se la mela tocca il bordo dello schermo il gioco finisce
 3. Se la mela tocca il giocatore ricompare poco sotto bordo superiore dello schermo e si sposta lungo l'asse y fino ad arrivare al bordo inferiore dello schermo
2. Il giocatore deve riuscire a prendere la mela, muovendosi con le frecce direzionali del computer
 1. Il giocatore si sposta a destra e sinistra (sull'asse x) utilizzando le frecce del computer
 1. Se il tasto freccia dx è premuto allora il gatto si sposterà verso destra
 2. Se il tasto freccia sx è premuto allora il gatto si sposterà verso sinistra
 3. Se il giocatore prende la mela, guadagna un punto.
 4. Se il giocatore guadagna 10 punti ha vinto e il gioco termina.

Qui sotto, trovi un po' di blocchi che potrebbero esserti utili per sviluppare il gioco.

