

# ROBOTICA EDUCATIVA CON IL KIT LEGO® EV3

## Comandare il nostro Robot!

Come fa il nostro robot a sapere che lavoro deve fare? Facile! Glielo diciamo noi! Più concretamente, siamo noi a dare delle istruzioni al robot che, eseguite correttamente e nell'ordine da noi stabilito, portano al completamento del suo compito. L'insieme dei comandi e delle istruzioni utili per il raggiungimento di un determinato scopo si chiama Algoritmo<sup>1</sup>.

Tuttavia, non basta soltanto pensare all'algoritmo che il nostro robot deve eseguire, ma è necessario comunicarglielo in un linguaggio comprensibile per il suo "cervello" (cioè, il suo computer).

Il problema è che i computer non "parlano" né italiano, né inglese, né nessuna lingua umana, loro capiscono solo il linguaggio macchina (composto da 0 e 1) che però per noi è molto difficile! Per risolvere questo problema, gli ingegneri hanno inventato i linguaggi di programmazione, che sono dei linguaggi con una difficoltà che si trova a metà tra quella dei linguaggi naturali (quelli umani) e quella dei linguaggi macchina.

Il processo di scrittura di un algoritmo in un linguaggio di programmazione, si chiama programmare. Nel caso dei robot LEGO® EV3, useremo un linguaggio di programmazione grafico.

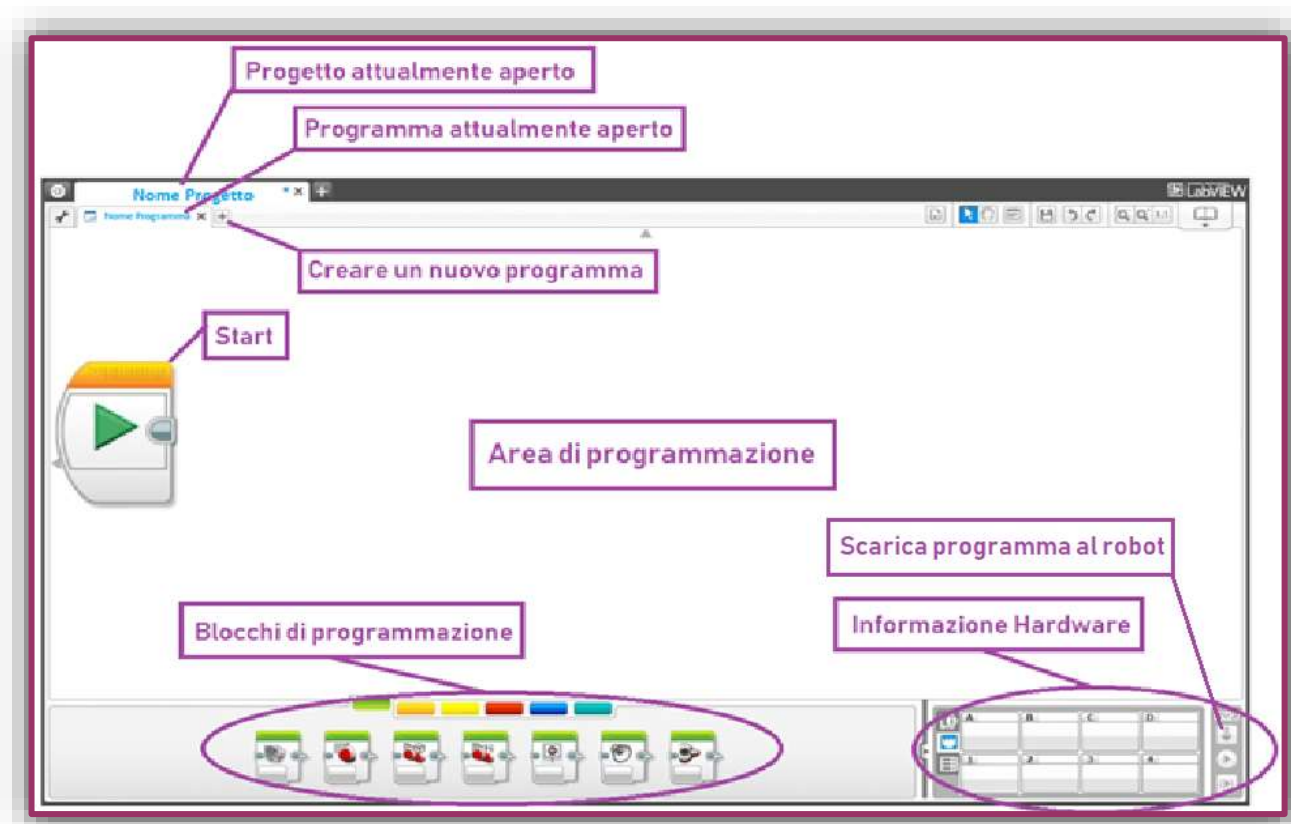
---

<sup>1</sup> Il termine Algoritmo deriva dalla trascrizione latina del nome del matematico persiano al-Khwarizmi vissuto nel IX secolo D.C., e considerato uno dei primi autori ad aver fatto riferimento a questo concetto scrivendo il libro "Regole di ripristino e riduzione".



## Interfaccia di Programmazione dei Robot LEGO® EV3

Per programmare i robot LEGO® EV3 c'è bisogno di un computer o di un tablet per poter installare l'interfaccia grafica di programmazione EV3. Questa ci consente di trascinare dei blocchi di programmazione all'interno di un'area di programmazione. Inoltre, una volta collegato il robot al computer (o al tablet), è anche possibile caricare il programma sul robot, cliccando la freccia in giù nella parte inferiore destra dell'interfaccia EV3. In questo modo, il robot potrà seguire le nostre istruzioni e fare i lavori e i compiti che noi vogliamo!



Interfaccia di Programmazione EV3

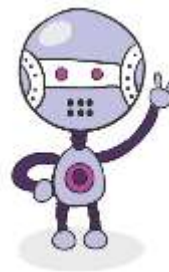


## I blocchi di programmazione

Il linguaggio di programmazione dei robot LEGO® EV3 è di tipo grafico, cioè vuol dire che non c'è bisogno di scrivere le istruzioni del nostro algoritmo (al contrario di altri linguaggi come Python, Java o C), ma è sufficiente collegare dei blocchi che le rappresentino.

Ci sono le seguenti 6 categorie di blocchi:

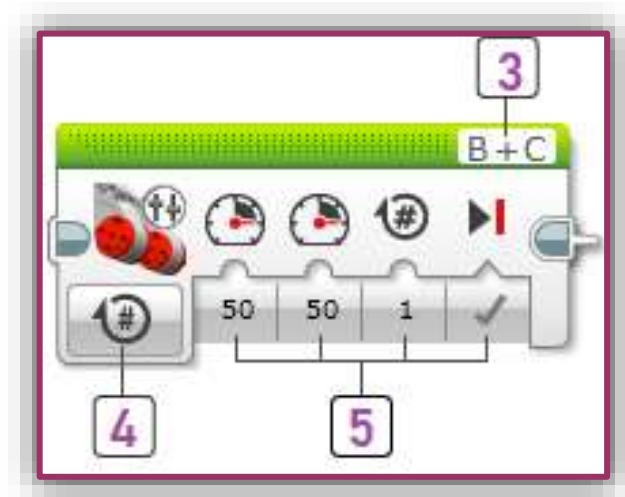
- Blocchi di **Azione**
- Blocchi di **Controllo di Flusso**
- Blocchi di **Sensori**
- Blocchi di **Dati**
- Blocchi **Avanzati**
- Blocchi **Personalizzati**



## Missione 1: Muoviti Robot!

Istruzioni:

1. Trascina all'interno dell'Area di Programmazione il blocco "Move Tank"
2. Collega il blocco appena trascinato al blocco di Start
3. Seleziona i motori giusti da muovere all'interno del blocco Move Tank
4. Seleziona un tipo di movimento:
  - a. Continuo
  - b. Per un certo numero di secondi
  - c. Per un certo numero di gradi
  - d. Per un certo numero di rotazioni
5. Seleziona la potenza dei motori
6. Collega il Robot acceso al PC
7. Scarica il programma al Robot



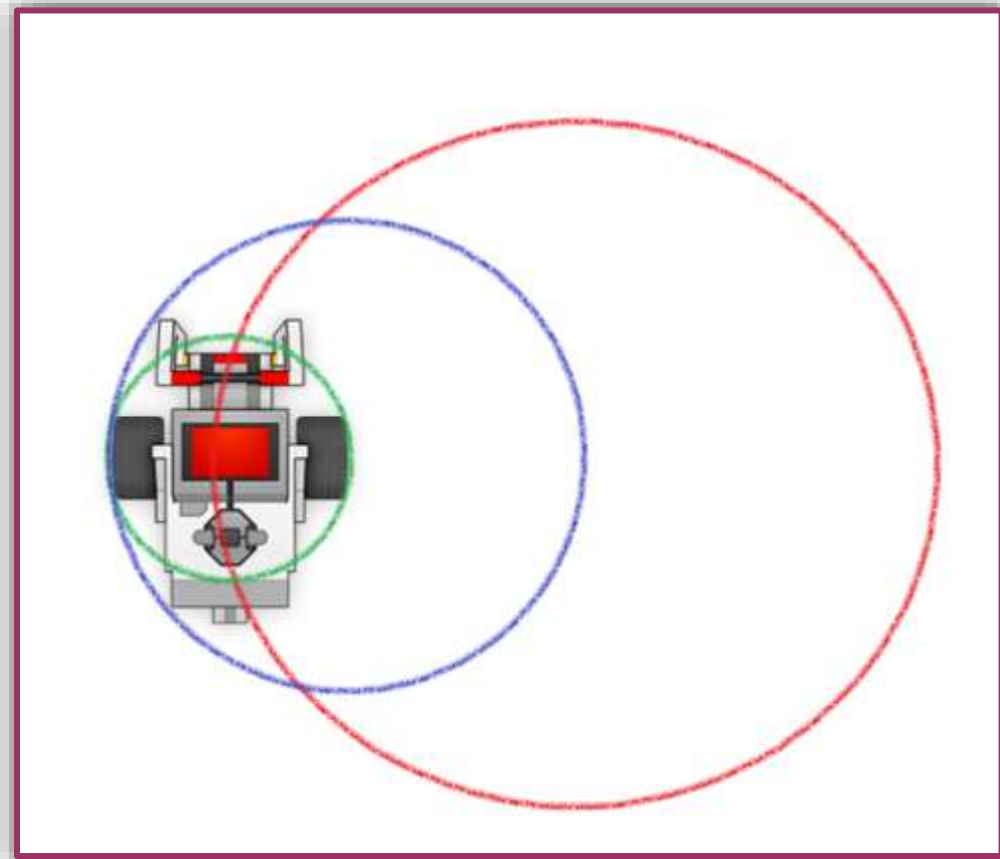
8. Esegui il programma nel Robot e guarda come si muove!

Blocco Move Tank

## Missione 2: Girati Robot!

*Cosa succede se la potenza dei motori del robot non è la stessa? In altre parole, cosa succede se un motore si muove ad una velocità diversa dell'altro?*

In questa missione impareremo come fare girare i nostri robot. Un robot può fare 3 tipi di giri:



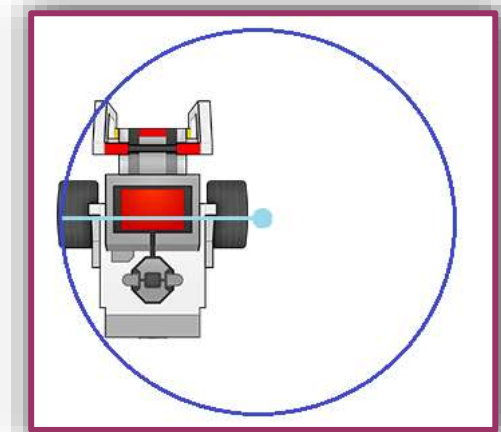
Giri del Robot



Il primo è il più semplice dei giri che può fare un robot e si chiama “Gradual Turn”. Consiste in un giro ampio nel quale entrambe le ruote del veicolo robotico si muovono nello stesso senso (a diversa velocità); quanto più piccola è la differenza di velocità fra i due motori delle due ruote, tanto più largo e ampio sarà il giro.

Il secondo tipo di giro si chiama “Swing Turn”, e consiste in un giro nel quale una delle due ruote rimane ferma, cioè la velocità del suo motore uguale a zero. Questo giro descrive un cerchio con un raggio uguale alla distanza fra le due ruote del robot.

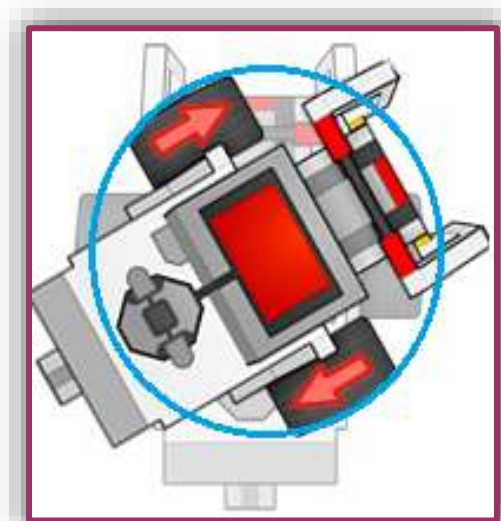
- SFIDA: Riesci a girare il robot di 180° ?



Swing Turn

Il terzo tipo di giro si chiama “Point Turn” e consiste in un giro nel quale entrambe le ruote del robot si muovono *in sensi opposti* e a diversa velocità. Questo giro descrive un cerchio più piccolo rispetto agli altri giri.

*Quale è il raggio del cerchio che descrive il Point Turn?*



Point Turn

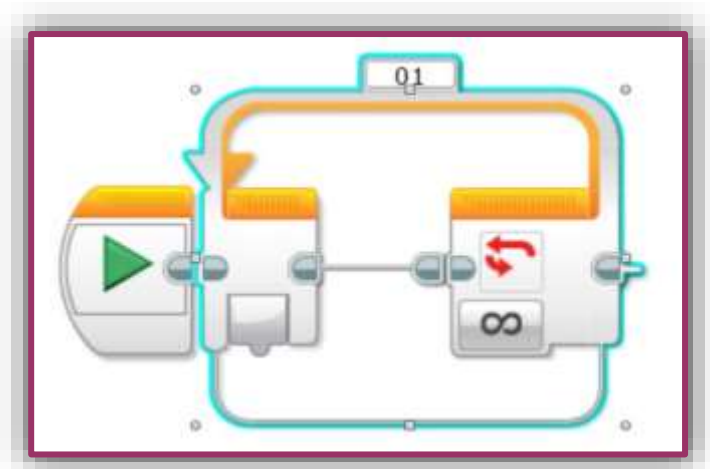
- SFIDA: Muovi il tuo robot descrivendo un quadrato di lato 1 m.

## Missione 3: Ripetendo con i Loop!

Quando si progetta un algoritmo, a volte, è necessario ripetere una stessa azione per un certo numero di volte. Per fare ciò, potresti pensare di mettere tanti blocchi identici nel tuo programma, ma questo non è il modo migliore per ripetere un'azione perché non solo è noioso, ma consuma memoria, risorse e può portare a errori.

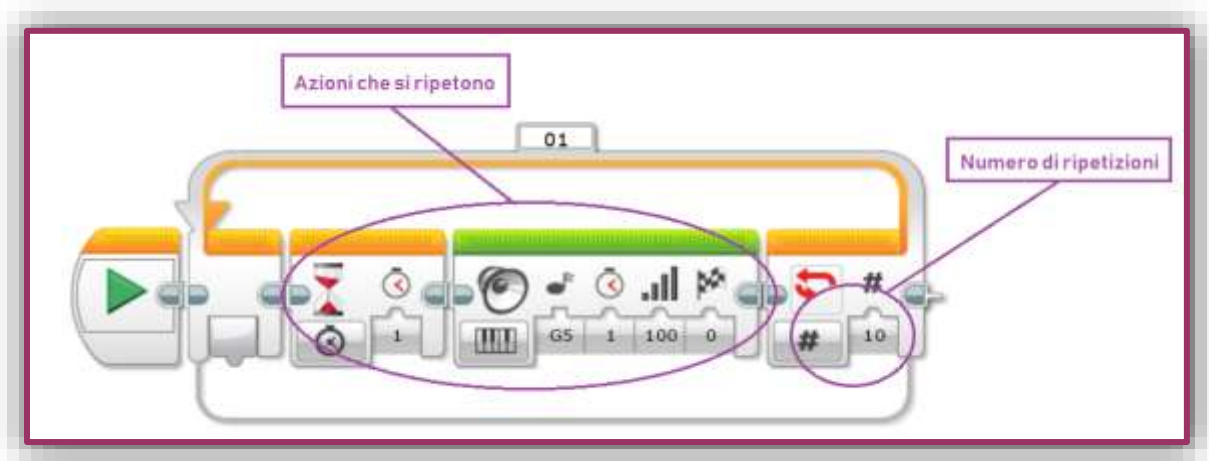
Il modo migliore per ripetere un'azione o una serie di azioni è usando il loop, che non è altro che un blocco specializzato che ripete altri comandi per un certo numero di volte.

Il loop è un blocco che assomiglia una cornice al cui interno si possono inserire altri blocchi di istruzioni. Tutto quello che si trova dentro la cornice verrà ripetuto per un certo numero di tempo (loop basato su tempo) o per un certo numero di volte (loop basato su un contatore) o finché non accadano certe condizioni (loop basato su condizioni).



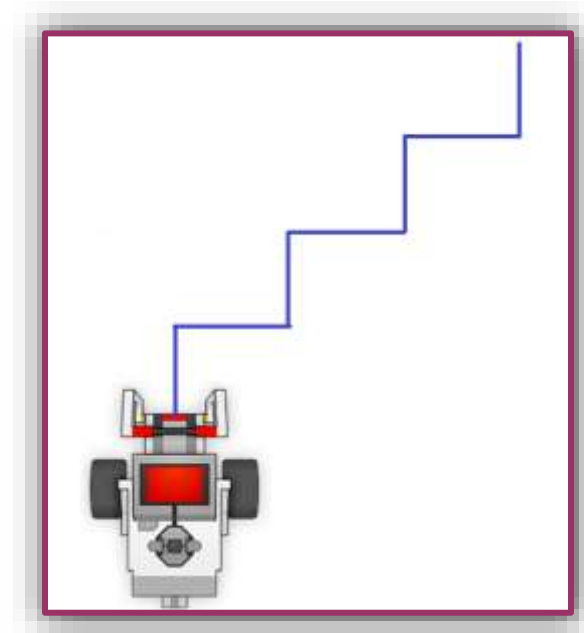
Blocco Loop

Esegui il seguente programma per riprodurre un suono dieci volte:



Esempio di Loop: Suono ripetuto 10 volte

- SFIDA: Usando il blocco Loop, programma il tuo robot per fare il seguente percorso:



Percorso per la SFIDA LOOP

- SFIDA: Usando il blocco Loop, programma il tuo robot per girare sul posto (point turn) *per sempre*.



## Missione 4: Decidendo con gli Switch!

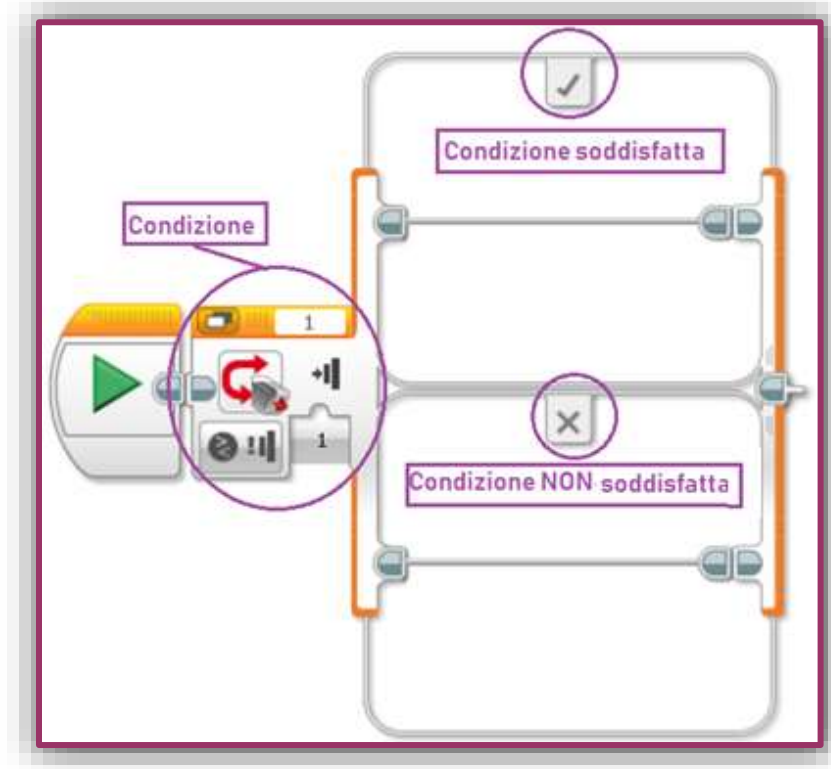
Facciamo decisioni tutti i giorni, anche senza rendercene conto! Per esempio, se vediamo il semaforo rosso, allora ci fermiamo. Se ho fame, allora mangio qualcosa. Il nostro cervello si fa costantemente domande del tipo: “*Se X condizione è vera, allora faccio una cosa. Altrimenti, ne faccio un'altra*”.

Questo tipo di domande “Se ... Allora” che ci permettono di fare decisioni, in programmazione si chiamano condizionali. I condizionali sono molto utili perché permettono anche al cervello del robot di porsi delle domande e di decidere, in base alla loro risposta, le azioni da intraprendere. Per esempio, se il nostro robot sta andando avanti, sarebbe molto utile fargli chiedere ogni tanto se c'è un ostacolo. In questo modo: “*Se c'è un ostacolo davanti, allora fermati. Altrimenti, continua ad andare avanti*”. Le domande che il robot si pone devono essere sempre a risposta chiusa: cioè SI o NO.



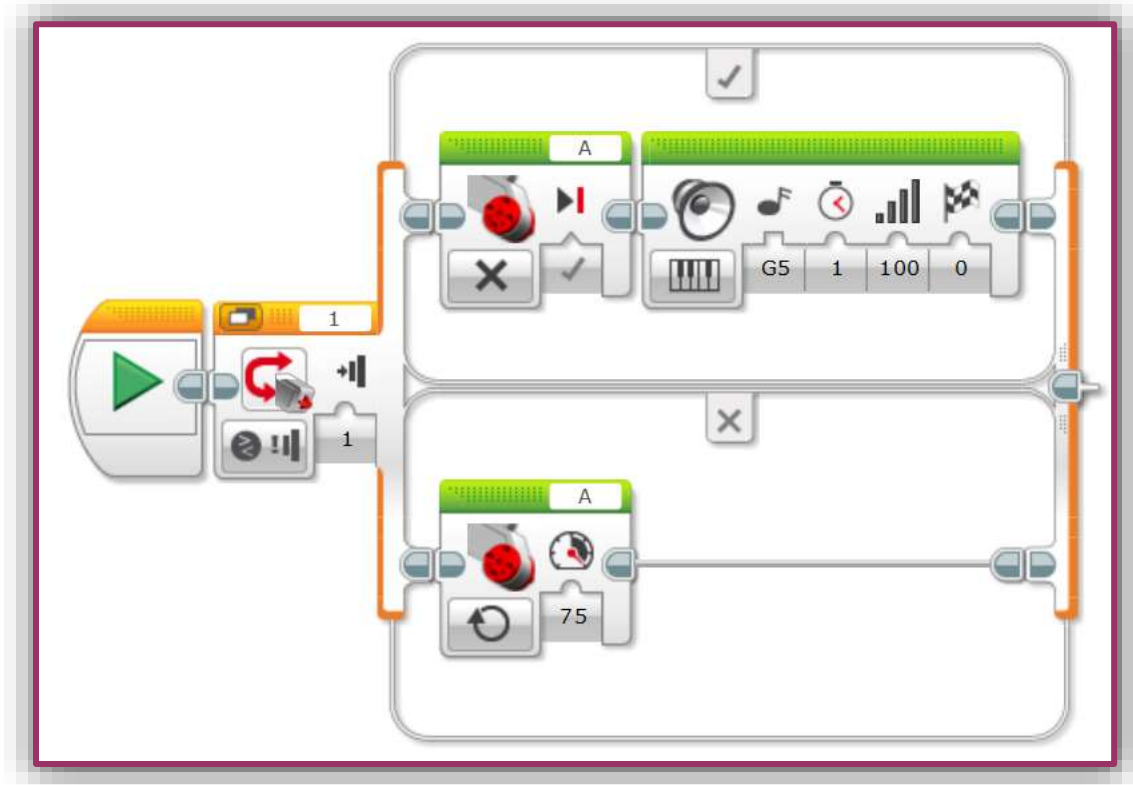


Il linguaggio di programmazione LEGO® EV3 ci permette di usare i condizionali tramite un blocco chiamato Switch. Questo blocco è composto da una condizione e da due scatole, che possono contenere altri blocchi di programmazione al loro interno. La scatola superiore (segnata da un tick) contiene le istruzioni che devono essere eseguite se la condizione viene soddisfatta (risposta positiva), mentre la scatola inferiore (segnata da una X) contiene le istruzioni che devono essere eseguite nel caso contrario (quando la condizione non è vera).



Blocco Switch

Vediamo in azione il blocco Switch mediante il seguente esempio:



Esempio di Switch

Nell'esempio precedente il cervello del robot si pone questo condizionale: "Se qualcuno preme il sensore di contatto, allora fermo il motore A e riproduco un suono. Altrimenti, muovo il motore A". In altre parole, questo programma muove il motore se nessuno tocca il sensore di contatto, perché se lo si tocca il motore si ferma e il robot emette un suono.

- SFIDA: Riesci a fare in modo che il robot abbia il comportamento dell'esempio precedente *per sempre*?
- SFIDA: Riesci a fare in modo che il robot abbia il comportamento dell'esempio precedente quando il *sensore di ultrasuono* rileva un ostacolo a meno di 50cm?

