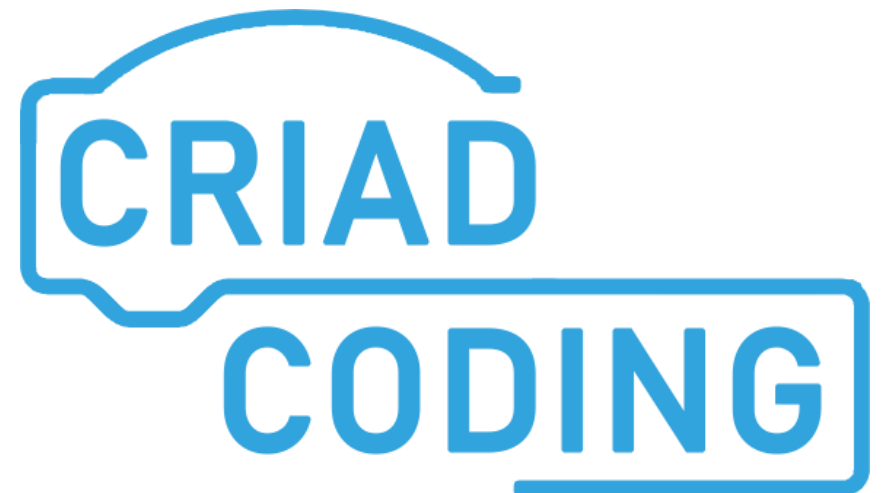


T2C – Training to Code

Input – Se - Variabili

Laura Tarsitano



Input

Quando si vuole interagire con il “pubblico” del micromondo possiamo fermare il flusso del programma e aspettare la risposta dell’utente.

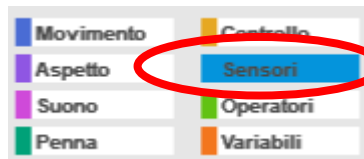
Per farlo usiamo il blocco dell’input che si trova nella categoria ‘Sensori’

Input

Possiamo creare un programma che richieda un determinato dato all'utente, che richieda quindi l'interazione con il "pubblico".

Il dato può essere di qualunque tipo: nome, età, numero da indovinare, ecc...

Per farlo utilizziamo il blocco dell'input, nella categoria 'Sensori'



Messaggio che vogliamo mostrare all'utente



La risposta fornita dall'utente viene memorizzata in una variabile predefinita già esistente chiamata 'risposta' che troviamo nella categoria 'Sensori'

risposta

Il blocco "chiedi" blocca il flusso del programma aspettando la risposta dell'utente. Riparte solo quando viene fornita la risposta.

Se

Quando ci troviamo davanti ad una condizione da valutare la domanda che ci si pone è:

“Se questa condizione è vera, allora faccio una cosa, altrimenti ne faccio un'altra”.

Questa domanda si traduce con il blocco del ‘Se’ che si trova nella categoria ‘Controllo’

Se

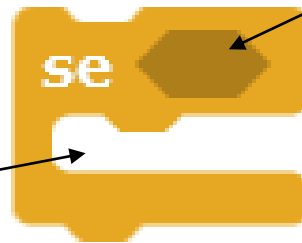
Quando dobbiamo prendere delle decisioni o ci troviamo davanti ad una scelta utilizziamo il blocco 'Se'.

Si trova nella categoria 'Controllo'.

Movimento	Controllo
Aspetto	Sensori
Suono	Operatori
Penna	Variabili

Se "la condizione è vera" →

Allora "fa quello che è scritto qui" →



Condizione

Esempio:
"Se sta squillando il telefono
allora rispondo".

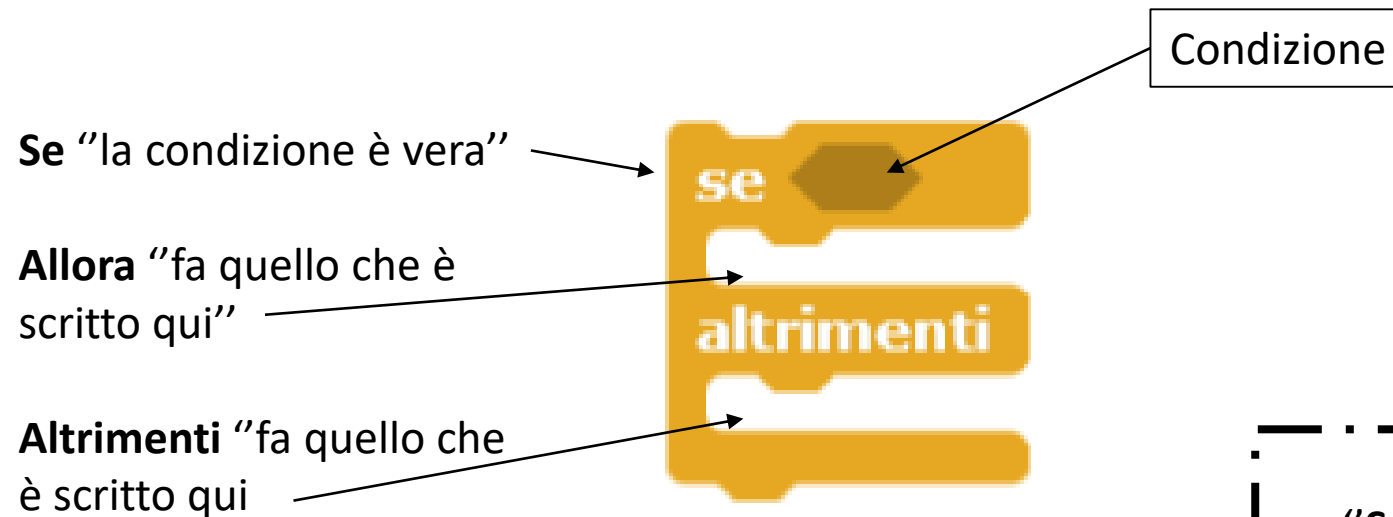
Se Altrimenti

Esiste un altro tipo di blocco 'Se'. È il 'Se Altrimenti'

Si trova nella categoria 'Controllo'

Sceglie una delle due strade in base alla valutazione della condizione.

NOTA: Sceglie solo una delle due strade, non le farà mai entrambe.



Esempio:
"Se il semaforo è verde
allora passo
altrimenti mi fermo".

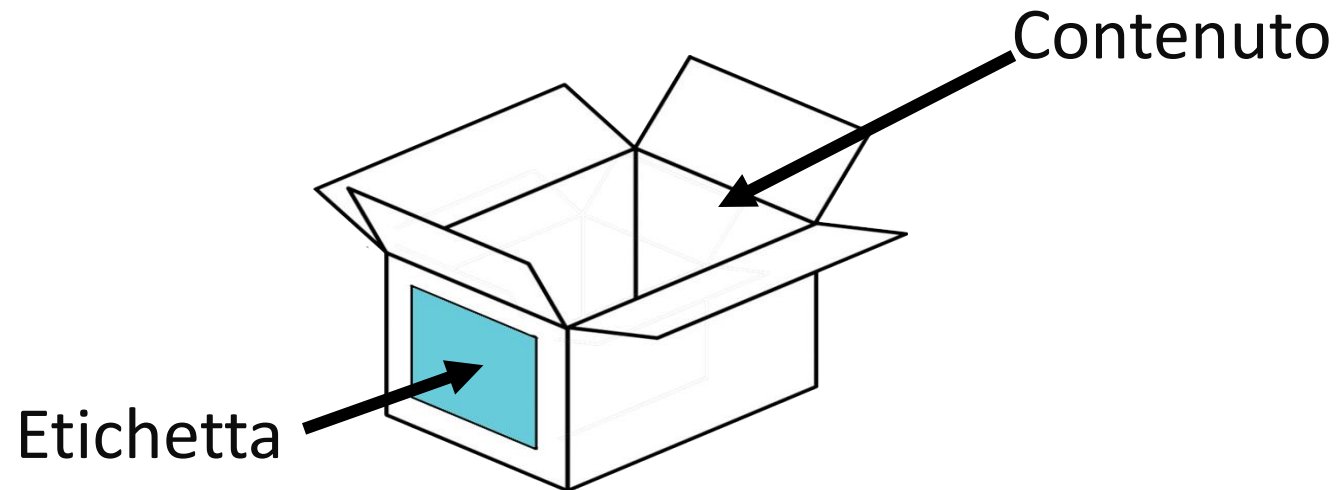
Variabili

Una variabile è un contenitore all'interno del quale possiamo memorizzare delle informazioni.

Variabili

Possiamo immaginare una variabile come una scatola, un contenitore, all'interno del quale inserire un'informazione da memorizzare.

Ogni variabile ha un **etichetta** che ne indica il nome e un **contenuto** che corrisponde al valore che inseriamo al suo interno (può essere di tipo numerico o alfanumerico).



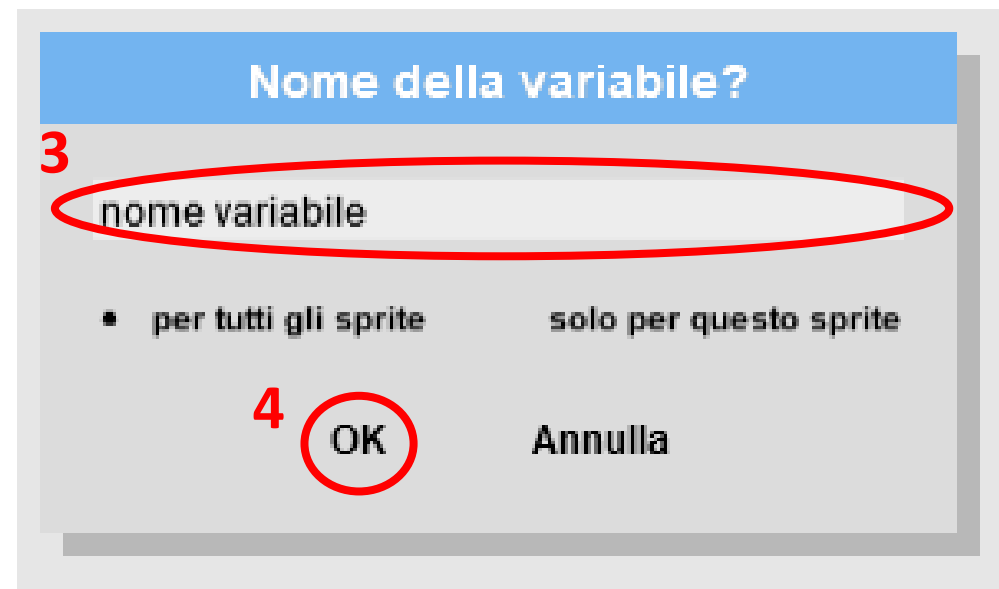
Variabili in Snap.

Come crearle, modificarle e cancellarle

Creare una nuova variabile - 1

Prova su
Snap!

Per creare una nuova variabile selezionare la categoria “Variabili” (1), scegliere “Nuova Variabile” (2), scrivere il nome da assegnare alla variabile (3) e fare click su OK (4)



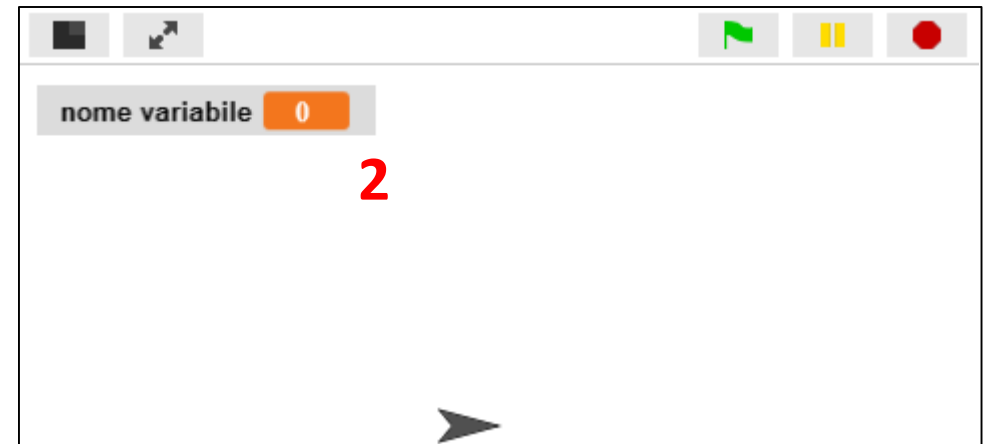
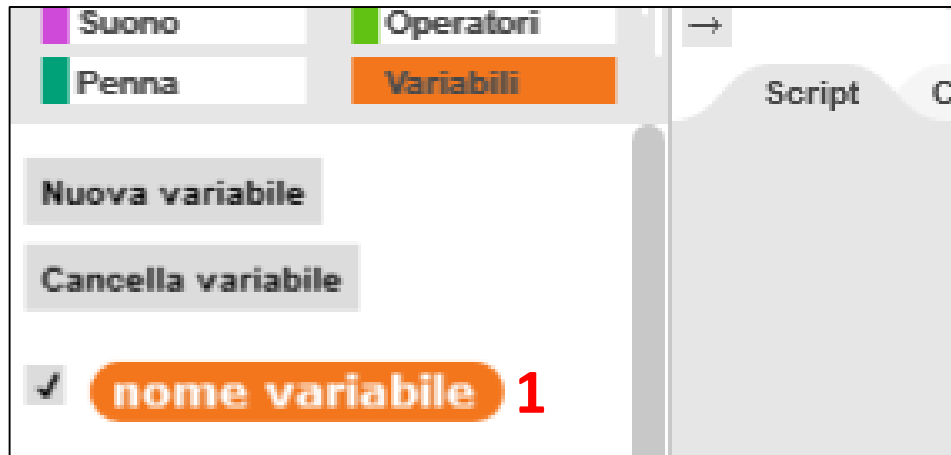
Creare una nuova variabile - 2

Prova su
Snap!

Una volta creata, la nuova variabile verrà visualizzata nella parte sinistra dello schermo (1).

Nello Stage verrà visualizzato il nome e il valore della variabile(2).

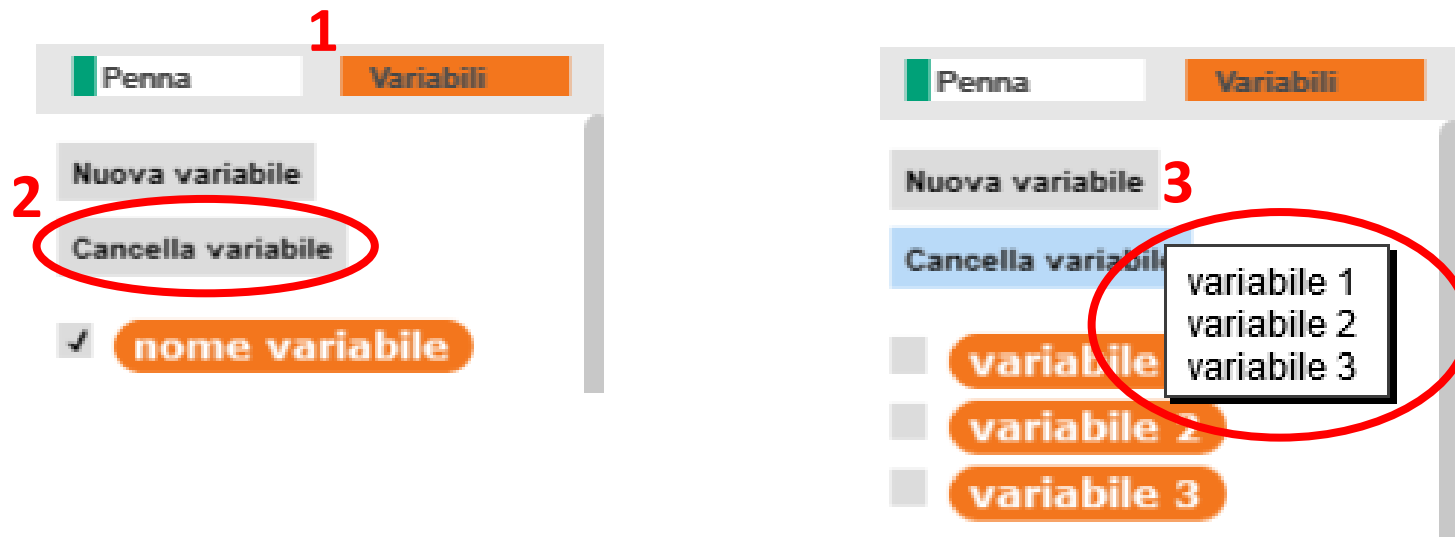
Per nascondere la variabile dallo Stage basterà togliere il segno di spunta accanto al nome della variabile (3).



Cancellare una variabile

Prova su
Snap!

Per cancellare una variabile dalla categoria “Variabili”(1) selezionare “Cancella Variabile”(2) quindi scegliere dal menù a tendina la variabile da eliminare(3)

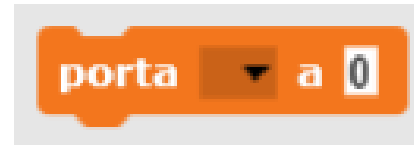


Assegnare un valore ad una variabile

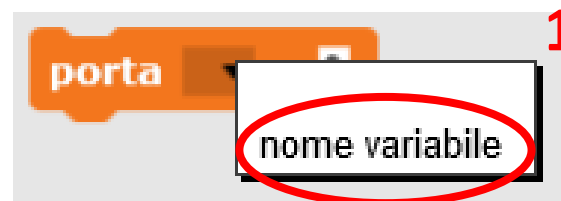
Prova su Snap!

Appena creata la variabile conterrà il valore 0.

Per assegnare un valore alla variabile (**riempirla**) usiamo il blocco "porta" nella categoria "Variabili".



Nel blocco porta selezionare la variabile a cui assegnare il valore dal menù a tendina(1) e indicare qual è il valore da assegnare(2).

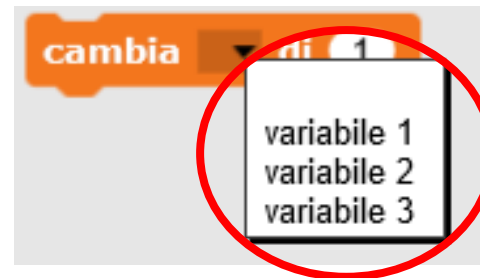


Cambiare il valore di una variabile - 1

Prova su
Snap!

Durante l'esecuzione di un programma il valore di una variabile potrebbe cambiare.

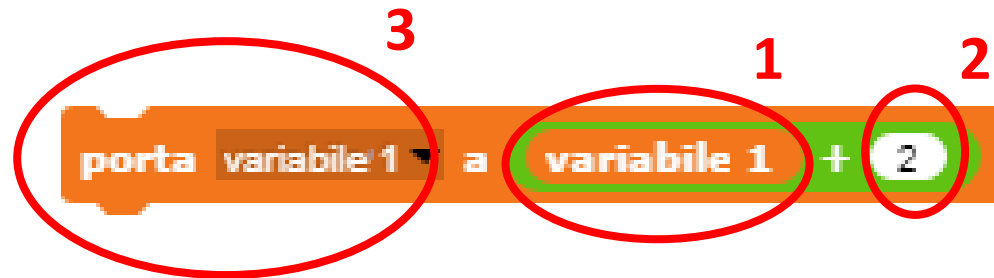
Per cambiare il valore di una variabile (sempre dello stesso valore) usare il blocco "cambia" nella categoria "Variabili", selezionare la variabile dal menù a tendina e indicare il valore.



Cambiare il valore di una variabile - 2

Prova su
Snap!

Un altro modo per cambiare il valore di una variabile è quello di ‘esplicitare’ l’operazione.



L’idea è: “controllo il valore della variabile (1), aggiungo un valore (2) e metto il nuovo valore nella variabile (3)”

Secondo Esempio - Idea

In questo Micromondo il protagonista estrae un numero a caso.

Il pubblico deve provare ad indovinare il numero estratto.

Secondo esempio - 1

Prova su
Snap!

Partiamo da un progetto vuoto per la costruzione del secondo esempio.

Primo obiettivo: *estrarre un numero casuale*

La prima cosa da fare è spiegare allo Sprite come fare ad estrarre un numero casuale e soprattutto **come e dove memorizzarlo**.

Bisogna sempre ricordare che lo Sprite non ricorda nulla, per far ricordare delle informazioni si devono utilizzare dei “supporti” aggiuntivi.

Questi supporti sono le **VARIABILI**

Secondo esempio – 2 – Creiamo la nuova variabile

Prova su
Snap!

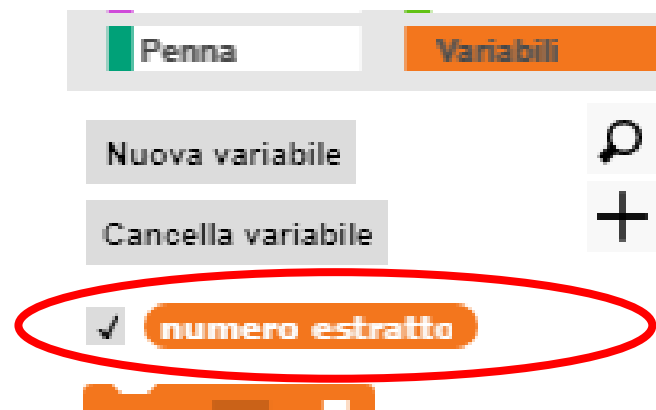
Primo obiettivo: *estrarre un numero casuale*

Ci serve una variabile per memorizzare il numero estratto.

Creiamo la variabile e la chiamiamo “**numero estratto**” (per la creazione di una variabile si rimanda alla slide [10](#)).

Una volta creata, vedremo la variabile nella categoria “Variabili”.

Il suo contenuto iniziale è 0.



Secondo esempio – 3 – Estraiamo un numero casuale

Prova su
Snap!

Primo obiettivo: *estrarre un numero casuale*

Il prossimo passo sarà quello di estrarre un numero casuale e di memorizzarlo all'interno della variabile.

Esiste un blocco che estrae un numero casuale tra un intervallo di numeri, questo blocco si trova all'interno della categoria 'Operatori'

numero a caso tra 1 e 10

Possiamo indicare l'intervallo di numeri tra i quali estrarre il numero

Dove memorizziamo il numero estratto? All'interno della variabile *numero estratto*.

Riempiamo la variabile *numero estratto* con il numero casuale.

porta numero estratto a numero a caso tra 1 e 10

Clicchiamo sulla bandierina verde per provarne il funzionamento.

Secondo esempio – Fine step 1

Prova su Snap!

Alla fine del primo obiettivo lo script si presenta così:



Secondo esempio – 4 – Indovina il numero

Prova su
Snap!

Secondo obiettivo: tentare di indovinare il numero

Il prossimo passo sarà quello di interagire col “pubblico”, ovvero con chi sta utilizzando il micromondo.

Per farlo utilizziamo il blocco dell’**input**, il blocco “chiedi” (per i dettagli del blocco dell’input si rimanda alla slide [3](#)).

chiedi e attendi

RICORDA: La risposta inserita dall’utente viene memorizzata nella variabile ‘risposta’ che si trova nella categoria Sensori.

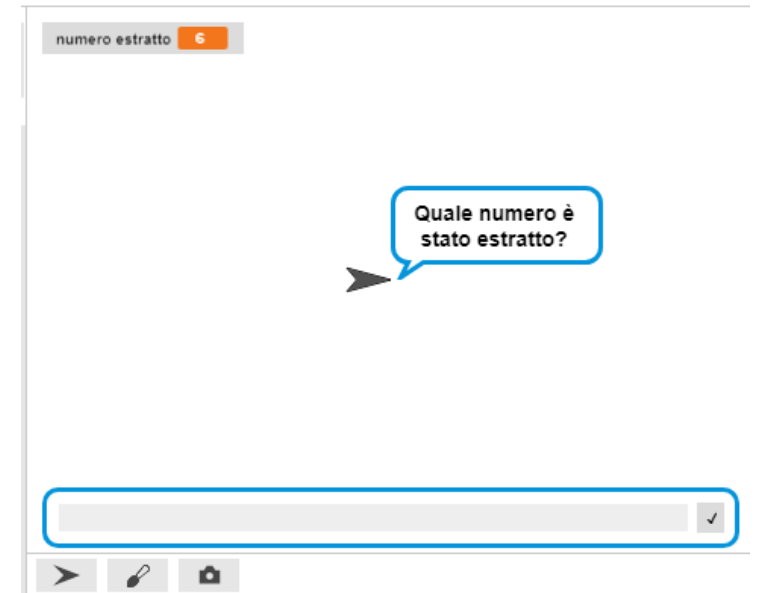
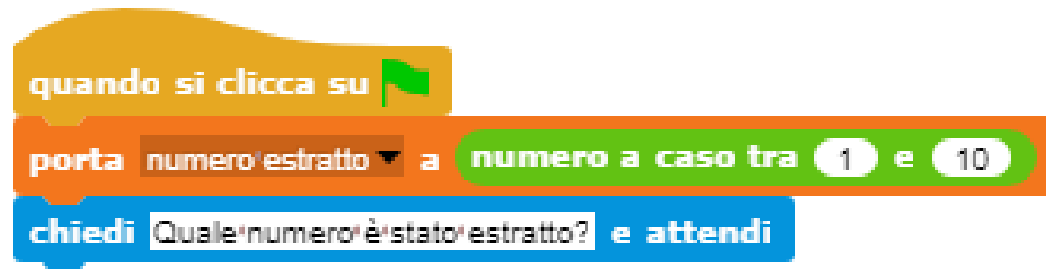
Per utilizzare il dato dovremo quindi recuperarlo dalla variabile risposta.

risposta

Secondo esempio – Fine step 2

Prova su
Snap!

Alla fine del secondo obiettivo lo script si presenta così:



Secondo esempio – 5 – Controlliamo la risposta

Prova su
Snap!

Terzo obiettivo: *controllare se si è indovinato il numero*

Il prossimo passo sarà **controllare** se la risposta inserita corrisponde al numero estratto.

Quando dobbiamo fare dei **controlli** utilizziamo il blocco del 'se' (per i dettagli si rimanda alla slide [5](#)).



Qual è la condizione da controllare?

*“La **risposta** inserita dall’utente è **uguale** al **numero estratto**?”*



Secondo esempio – 6 – Controlliamo la risposta

Prova su Snap!

Terzo obiettivo: *controllare se si è indovinato il numero*
"Se la risposta è uguale al numero estratto....."

```
se risposta = numero estratto
```

Allora l'utente ha indovinato"

```
dire Bravo! Hai indovinato! per 2 secondi
```

"Se la risposta è uguale al numero estratto
Allora l'utente ha indovinato"



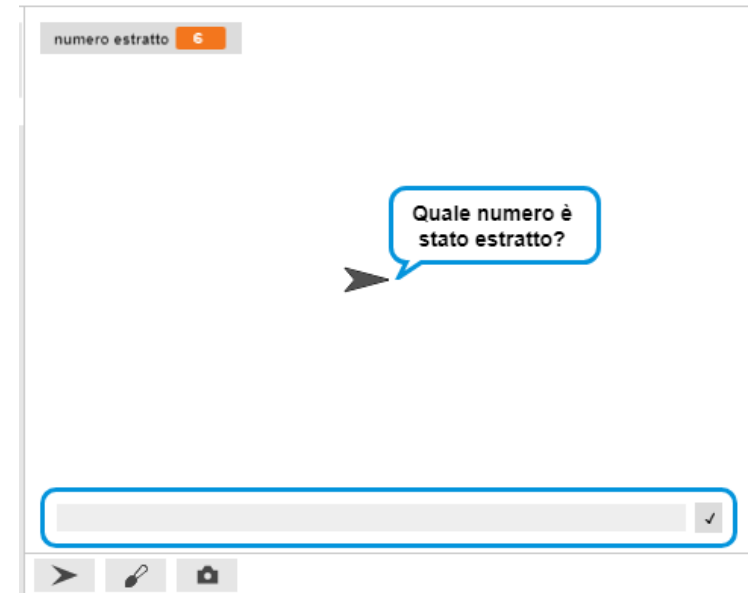
```
se risposta = numero estratto  
  dire Bravo! Hai indovinato! per 2 secondi
```


Secondo esempio – Fine step 3

Prova su
Snap!

Alla fine del terzo obiettivo lo script si presenta così:

```
quando si clicca su [bandierina]
  porta numero estratto a numero a caso tra 1 e 10
  chiedi Quale numero è stato estratto? e attendi
  se risposta = numero estratto
    dire Bravo! Hai indovinato! per 2 secondi
```



Secondo esempio – 7 – Altrimenti...?

Prova su
Snap!

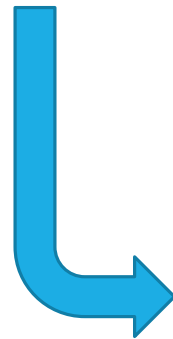
Quarto obiettivo: *inseriamo la risposta per il numero sbagliato*

“Se la risposta è uguale al numero estratto

Allora l’utente ha indovinato

Altrimenti l’utente ha sbagliato”

In questo caso ci servirà un “se altrimenti”



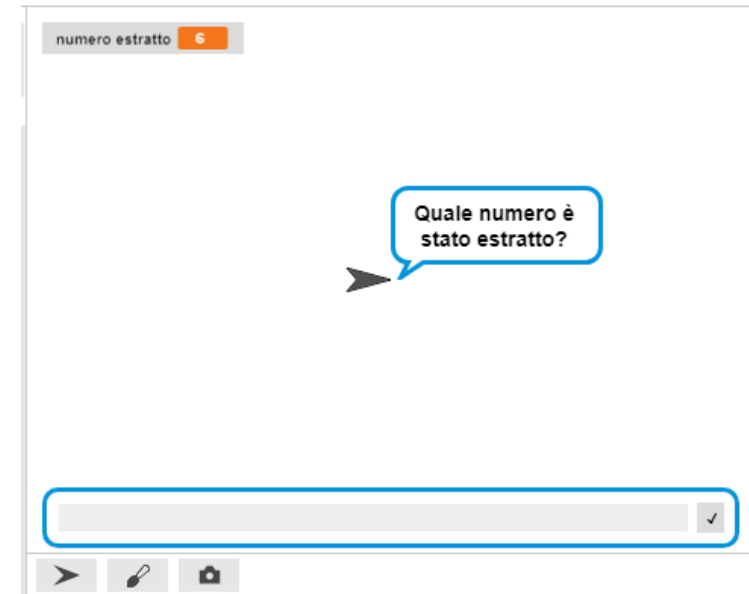
```
se risposta = numero estratto
  dire Bravo! Hai indovinato! per 2 secondi
altrimenti
  dire Pecato! Hai sbagliato! per 2 secondi
```

Secondo esempio – Fine step 4

Prova su Snap!

Alla fine del quarto obiettivo lo script si presenta così:

```
quando si clicca su
  porta numero estratto a numero a caso tra 1 e 10
  chiedi Quale numero è stato estratto? e attendi
  se risposta = numero estratto
    dire Bravo! Hai indovinato! per 2 secondi
  altrimenti
    dire Peccato! Hai sbagliato! per 2 secondi
```



Secondo esempio – Fine, ma...

Il Micromondo potrebbe già essere terminato così.

Vogliamo però aggiungere due dettagli per renderlo più completo.

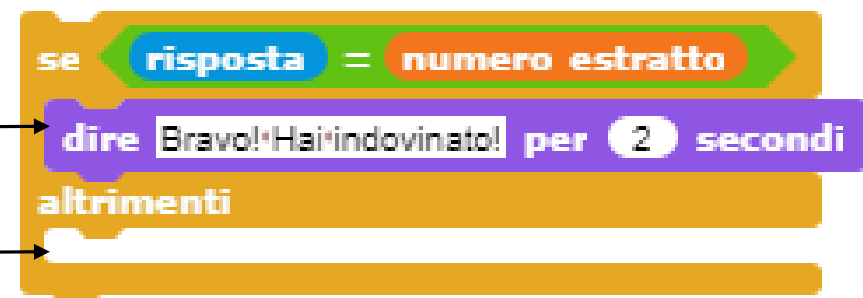
Se **la risposta è corretta** il programma termina dicendo “Bravo hai indovinato!”

Ma nel caso la risposta fosse **sbagliata (altrimenti)** vogliamo:

1. Che il personaggio ci dica se il numero inserito è più grande o più piccolo del numero estratto
2. Riprovare a indovinare

La parte relativa alla risposta corretta è OK

Dobbiamo lavorare sulla parte della risposta sbagliata



Secondo esempio – 8 – Più grande o più piccolo?

Prova su Snap!

Primo obiettivo: capire se il numero inserito sia più grande o più piccolo del numero da indovinare

Siamo nel caso dell'altrimenti

Dobbiamo farci un'altra domanda:

**“Se la risposta è maggiore del numero estratto
Allora il numero da indovinare è più piccolo
Altrimenti il numero da indovinare è più grande”**

Traduciamolo con i blocchi:

```
se risposta = numero estratto  
  dire Bravo! Hai indovinato! per 2 secondi  
altrimenti
```

```
se risposta > numero estratto  
  dire Il numero da indovinare è più piccolo per 2 secondi  
altrimenti  
  dire Il numero da indovinare è più grande per 2 secondi
```

Inseriamo ora questo se all'interno dell'altrimenti

Secondo esempio – Fine step

Prova su Snap!

Alla fine di questo primo step lo script si presenta così:

```
quando si clicca su
  porta numero estratto a numero a caso tra 1 e 10
  chiedi Quale numero è stato estratto? e attendi
  se risposta = numero estratto
    dire Bravo! Hai indovinato! per 2 secondi
  altrimenti
    se risposta > numero estratto
      dire Il numero da indovinare è più piccolo per 2 secondi
    altrimenti
      dire Il numero da indovinare è più grande per 2 secondi
```



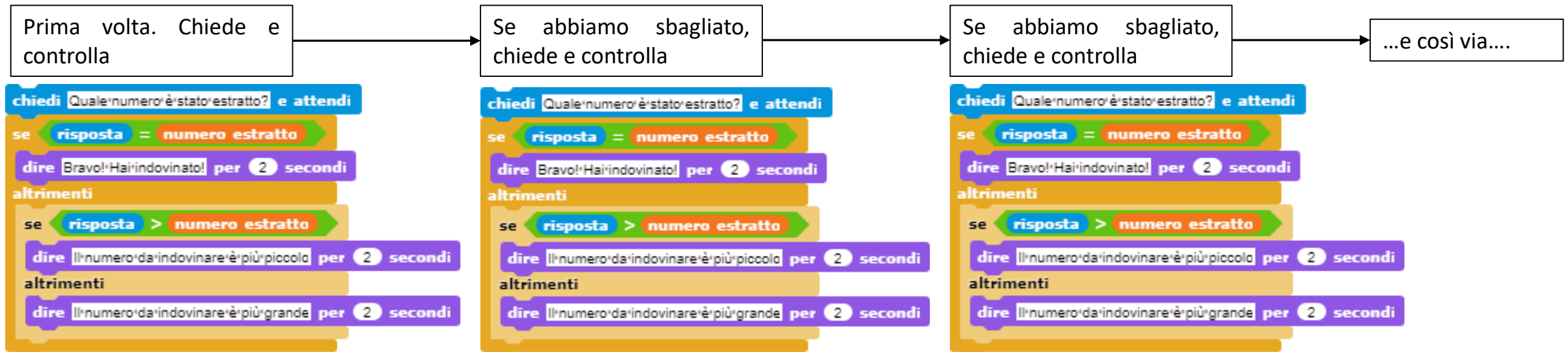
Secondo esempio – 9 – Ritenta

Secondo obiettivo: *Ritentare a indovinare il numero*

Una volta fatto il controllo, vogliamo riprovare a indovinare il numero.

Per farlo dovremmo re-inserire la risposta. Il blocco che ci permette di inserire la risposta è il blocco 'chiedi'. Poi dovremmo ri-controllare la risposta, se è giusta Ok, altrimenti dovremmo ri-tentare e così via...

Tradotto in blocchi vuol dire:



Secondo esempio – 10 – Ripetiamo fin quando.....

Prova su
Snap!

Secondo obiettivo: *Ritentare a indovinare il numero*

Notiamo come dobbiamo ripetere per un certo numero di volte sempre gli stessi blocchi. Ci serve un ripeti!

Ma che tipo di ripeti ci serve? Fino a quando dobbiamo ripetere?

Ci serve un “Ripeti fino a quando”



ripeti fino a quando

Possiamo decidere noi fino a quando deve ripetere.

Deve ripetere fin quando non indoviniamo il numero. Cioè fino a quando la risposta è uguale al numero estratto:



risposta = numero estratto

Secondo esempio – Fine

Prova su
Snap!

Alla fine della costruzione del Micromondo lo script si presenta così:

```
quando si clicca su [bandierina verde]
  porta numero estratto a numero a caso tra 1 e 10
  ripeti fino a quando [risposta = numero estratto]
    chiedi [Quale numero è stato estratto?] e attendi
    se [risposta = numero estratto]
      dire [Bravo! Hai indovinato!] per 2 secondi
    altrimenti
      se [risposta > numero estratto]
        dire [Il numero da indovinare è più piccolo] per 2 secondi
      altrimenti
        dire [Il numero da indovinare è più grande] per 2 secondi
```